Universidad de Valencia
Departamento de Ingeniería Electrónica

Design, development and implementation of a readout system for microstrip silicon sensors. Upgrade for test beam measurements

TESIS DOCTORAL

Ricardo Marco Hernández

2012

El **Dr. Salvador Martí i García**, Investigador Científico del CSIC, y el **Dr. Carlos Lacasta Llácer**, Investigador Científico del CSIC,

CERTIFICAN:

que la presente memoria, **"Design, development and implementation of a readout system for microstrip silicon sensors. Upgrade for test beam measurements"**, ha sido realizada bajo nuestra dirección en el Departamento de Ingeniería Electrónica de la Universidad de Valencia por Ricardo Marco Hernández, teniendo como tutor al **Dr. Vicente González Millán**, profesor titular del Departamento de Ingeniería Electrónica; y constituye su tesis para optar al grado de doctor en Ingeniería Electrónica por la Universidad de Valencia.

Y para que conste, en cumplimiento de la legislación vigente, firmo el presente certificado en Burjassot, a 7 de mayo de 2012.

Dr. Salvador Martí i García

Dr. Carlos Lacasta Llácer

Contents

Introduction

This thesis summarizes the work carried out for the design, development and implementation of a readout system for microstrip silicon sensors as well as its upgrade in order to include part of the system in a telescope for test beam measurements. The main properties of highly irradiated microstrip silicon sensors must be studied since a high luminosity is intended to be achieved at HL-LHC (High Luminosity Large Hadron Collider) experiments. In particular, the charge collected when a charged particle crosses the detector is important for the detector performance. It is difficult to carry out meaningful measurements. First, a custom and expensive laboratory set up is needed. Second, the measurements obtained with different setups could be not comparable if they are not calibrated correctly. Finally, this type of sensors may have hundreds of channels in order to read out. It would be also interesting to test this kind of detectors with an electronic system as similar as possible to those used at the LHC (Large Hadron Collider) experiments, so a front-end readout chip as those used at the LHC experiments should be used. Moreover, an analogue measurement of the front-end pulse shape is preferred over a binary one for charge collection research.

An electronic system which can acquire an analogue measurement has been developed. The system can be used with a laser setup, where a laser light is generated exciting a laser source with a pulsed signal. It can be used also with a radioactive source setup, where the charged particles are generated randomly.

Furthermore, the high resolution of silicon microstrip detectors depends on different parameters as the silicon detector characteristics, the readout pitch, the presence of intermediate floating strips, the coupling with the readout electronics or the noise level of the readout electronics. The spatial resolution of a new detector can be determined exactly by means of a detailed study in a test beam. The experimental conditions in a test beam facility are more similar to a high energy physics experiment than the conditions in the laboratory. Therefore, a telescope for the measurement of important parameters in new microstrip or pixel detectors in a beam test environment has been designed. Part of the former readout system developed for the laboratory environment has been upgraded in order to include it in this telescope.

The framework in which the system has been developed is accomplished in chapter 1. Thus, a particle physics overview is carried out. Then, a summarized description of the LHC experiments and a brief explanation of the main characteristics of the HL-LHC are accomplished. Finally, the RD50 collaboration as well as the ALIBAVA collaboration aims and structure are explained.

The silicon detectors operation principles are summarized in chapter 2. The silicon properties and the *pn* junction behaviour are described briefly. The main *pn* junction characteristics of interest for silicon detectors are explained. Moreover, an overview of the radiation damage effects is carried out. The structures used for producing silicon detectors are also presented. Subsequently, the front-end electronics stages used for the signal acquisition are described as well as the noise sources of the silicon sensors and the front-end electronics.

In chapter 3, the motivations for the design of the readout system for microstrip silicon sensors, its specifications and its architecture are detailed and discussed. A brief description of the experimental methods for determining the main parameters of the silicon sensors is carried out. In particular, the radioactive source setup and the laser setup are described. Then, a discussion about the advantages and the disadvantages of the current acquisition systems used with these setups is done. The motivations for designing the readout system are derived from this discussion as well as its specifications. Finally, the system architecture adopted in order to fulfill these specifications is described.

The chapters 4, 5 and 6 are devoted to the description of the different parts which compose the system. Therefore, chapter 4 is dedicated to the design of the daughter board, chapter 5 to the design of the mother board and chapter 6 to the software.

In chapter 7, the development of the system, its production as well as some improvements carried out in the system are described. The development process of the system to validate the design of the system is detailed. Measurements performed with the system are presented. These measurements were done both using a laser setup and a radioactive source setup. Different types of non-irradiated and irradiated microstrip silicon sensors were used to analyze the system performance. Then, the production process of the system and the quality tests carried out are explained. Finally, the improvements carried out in the system are detailed.

Chapter 8 is devoted to the upgrade of the mother board in order to include it as a part of a telescope for test beam measurements. The architecture of this telescope

and the motivations for developing it are explained. The different parts of the telescope are described. The upgrade of the mother board for the telescope is treated, with special emphasis on the FPGA logic and the embedded processor firmware upgrade. The development tests carried out to validate the mother board upgrade are also described.

Chapter 1

Framework

This chapter is an introduction to the work presented in this thesis. First, a brief overview of the particle physics is made in section 1.1. The main constituents of the matter and anti-matter are explained. Also, the fundamental forces present in the nature. The role of the CERN as a reference laboratory of particle physics is discussed in section 1.2. In section 1.3, the main characteristics of the Large Hadron Collider (LHC) and the LHC experiments are briefly described. Then, the possible upgrade for the LHC, so-called the High Luminosity LHC (HL-LHC), is treated in section 1.4. Finally, in section 1.5, the RD50 Collaboration and the ALIBAVA Collaboration are described and stated as the frame of this work.

1.1. Particle physics overview

The huge particle physics experiments studying tiny furtive particles have a common goal, to discover what matter is made of and how it holds together. Since everything is made of matter this leads to the understanding of the few very fundamental forces which make the world what it is. Nowadays, matter is found to be made from twelve basic building blocks called fundamental particles. These basic bricks are governed by four fundamental forces. The theory which describes these basic blocks and three of the fundamental forces is called the Standard Model [1] of particle physics. It has been developed during the 20th century and is about stable since 1974. Since then, particle physicists try to test the predictions of the Standard Model and to determine precisely its parameters. They try to find the limitations of the model and find new physics beyond the Standard Model.

It is known, at least since *Medeleev's* periodic classification of the elements, that all matter is made of atoms. *Rutherford* showed that the atom is made of a very hard and extremely small nucleus surrounded by a cloud of electrons. The nucleus is positively charged and the electrons negatively. The atom is bound together by the electric attraction of opposite charges. The total charge of the atom is usually neutral. The nucleus is made of protons and neutrons. They are the elementary bricks of the nucleus, the most of the mass of an atom. Protons have a positive electric charge while neutrons are neutral. The number of protons defines the charge of the nucleus, which defines the type of the atom. The nucleus holds together because of the Strong force. This force is called this way because it is much stronger than the electromagnetic force and it counteracts the electromagnetic repulsion of the protons. Protons, neutrons and electrons: this is what the whole world is made of. But these are not yet the fundamental bricks.

1.1.1. Matter

The fundamental matter constituents are the fermions, which are divided into quarks and leptons. The former are particles with a fractional electron charge and they experience the four fundamental forces. The latter have unitary charge and they do not experience all the fundamental forces. Each group consists of six particles which are related in pairs or generations. The lightest and most stable particles make up the first generation, whereas the heavier and less stable particles belong to the second and third generations. All stable matter is made from particles that belong to the first generation, any heavier particles quickly decay to the next most stable level. The six quarks are paired in three generations. The six leptons are similarly arranged in three generations. The electron, the muon and the tau all have an electric charge and a mass, whereas the neutrinos are electrically neutral with very little mass. A summary of the matter constituents is shown in table 1.1.

1.1.2. Anti-matter

For every particle type there is a corresponding anti-particle type. Each particle and the corresponding anti-particle have identical mass and spin but opposite charges. Therefore, there are six anti-quarks and six anti-leptons. As particles and anti-particles, matter and anti-matter have the same properties. Since electrons, neutrons and protons are stable their anti-particles are stable too. If one imagines a mirror which exchanges charges instead of sides, an anti-universe would be the reflection of the universe in this mirror. When anti-matter meets an equal quantity of matter it annihilates and produces energy in the form of light following

Einstein's equation:

$$E = m \cdot c^2 \quad (1.1)$$

where m is the total mass of matter and anti-matter involved and c the speed of light.

Fermions					
Quarks (spin 1/2)			Leptons (spin 1/2)		
Flavor	**Approximate mass (MeV/c²)**	**Electric Charge**	**Flavor**	**Mass (MeV/c²)**	**Electric Charge**
First generation					
Up (u)	2	2/3	***Electron (e-)***	0.511	-1
Down (d)	5	-1/3	***Electron-neutrino (ν_e)***	$<0.13\times10^{-6}$	0
Second generation					
Charm (c)	1300	2/3	***Muon (μ-)***	105.61	-1
Strange (s)	100	-1/3	***Muon-neutrino (ν_μ)***	$<0.13\times10^{-6}$	0
Third generation					
Top (t)	173000	2/3	***Tau (τ-)***	1777	-1
Bottom (b)	4200	-1/3	***Tau-neutrino (ν_τ)***	$<0.14\times10^{-6}$	0

Table 1.1. *Summary of fermions. Masses are given in MeV/c², following the equivalence E = mc². Electric charges are given in units of proton charge.*

1.1.3. Fundamental forces

There are four fundamental forces in nature (table 1.2): gravity, electromagnetism, weak nuclear force and strong nuclear force. They work over different ranges and have different strengths.

Gravity is the weakest force but it has an infinite range. It is not described by the Standard Model of elementary particles because of theoretical difficulties. Since it is such a weak force, the effect of gravity is so weak as to be negligible for tiny particles. The gravitational attraction of two objects is proportional to their

mass. It is thus only relevant for heavy objects.

Properties of the fundamental forces					
		Gravity	**Weak**	**Electromagnetic**	**Strong**
Acts on		Mass-Energy	Flavor	Electric Charge	Color Charge
Particles experiencing		All	Quarks, Leptons	Electrically Charged	Quarks, Gluons
Particles mediating		Graviton	$W^+ W^- Z^0$	γ	Gluons
Strength at	**10^{-18} m**	10^{-41}	0.8	1	25
	$3x10^{-17}$ m	10^{-41}	10^{-4}	1	60

Table 1.2. *Main properties of the fundamental forces. The strengths are shown relative to the strength of the electromagnetic force of two u quarks separated by the specified distances.*

Bosons					
	W boson (W^{+})	**W boson (W^{-})**	**Z boson (Z^{0})**	**Photon (γ)**	**Gluon (g)**
Force		Weak		Electromagnetic	Strong
Mass (GeV/c^2)	80.39	80.39	91.188	0	0
Electric charge	+1	-1	0	0	0
Spin	1	1	1	1	1

Table 1.3. *Summary of force carriers. Masses are given in GeV/c^2, following the equivalence $E = mc^2$. Electric charges are given in units of proton charge.*

The electromagnetic force also has infinite range but it is much stronger than gravity. It is the cause of the attraction or repulsion between electrically charged or magnetic objects. The electromagnetic force also guarantees the stability of atoms and molecules. Moreover, it is the source of light, since every emission of light is an electromagnetic process. The photon (γ) is a piece of light but also the mediator of the electromagnetic force. This means that when for instance two charged

objects, or two magnets attract (or repel) each other, they exchange photons.

The strong force is the strongest among all the four fundamental interactions within its short range. It binds quarks together and binds nucleons (protons and neutrons) together in the nucleus. This force keeps the nucleus stable as it is much stronger than the electromagnetic repulsion. The strong force charge is named colour. Every quark carries one of the following colours: red, green and blue but, due to the short range of this force, no coloured particle can occur in nature. This is the reason why the quarks are always hidden in hadrons. The baryons (nucleons, as the proton or the neutron, and hyperons, as the lambda, the sigma, the xi or the omega) are made up of three quarks carrying each of the three colours (the sum being white) and in the mesons (for instance, the pion, the kaon or the eta) the quark and the anti-quark carry the same colour, which counts negatively in the case of the anti-quark. Only quarks can be coloured and hence feel the strong force. The mediator is the gluon (g), a massless particle which exchanges the colours of two quarks.

The weak force is much stronger than gravity but it is indeed the weakest of the other three. It is also a short-range force. All particles feel the weak force. The weak force is the cause of the instability of the heavy quarks and leptons. The mediators are the W and the Z^0, two very heavy particles (almost a hundred times the mass of the proton) which were predicted by the Standard Model and discovered at CERN in the early eighties. For instance, the nuclear beta (β) decay is due to the weak force interaction. In this decay one of the neutrons of the nucleus decays into a proton, emitting and electron and a neutrino. Table 1.2 and table 1.3 summarize the properties of the fundamental forces and the bosons (force carrier particles), respectively.

1.1.4. The Higgs boson

The Standard Model has unified successfully two of the four fundamental forces, namely the weak force and the electromagnetic force, in the electroweak force. These two forces are described within the same theory, which forms the basis of the Standard Model. In order to validate this unification, it is required that the force-carrying particles have no mass but it is known that this is not true as the W and Z^0 are very heavy. Therefore, it must be explained why there are massive bosons.

A solution for this question is the Higgs boson (H^0). The Higgs boson would be the mediator of the Higgs field. Any particle that interacts with this field is given a mass via the Higgs boson. The more the particle interacts, the heavier it becomes,

whereas particles that never interact are left with no mass at all. The problem is that the Higgs boson has never been observed in an experiment to confirm the theory. Furthermore, the mass of the Higgs boson is not known, which makes it more difficult to identify. Therefore, it must be sought by systematically searching a range of mass within which it is predicted to exist. The yet unexplored range is accessible using the Large Hadron Collider, which would determine the existence of the Higgs boson.

1.1.5. Extensions of the Standard Model

The Standard Model is a theory which describes in a coherent way three fundamental forces (strong, weak and electromagnetic) and seems incompatible with the fourth (gravity). This is a good reason to think that it needs to be extended. There are other arguments going into the same direction. First, the Standard Model does not contain the necessary physics to explain the creation of the universe. Second, the Standard Model has seventeen parameters, which is felt to be too much for a fundamental theory. Newton's gravity has one parameter, Maxwell's electromagnetism has three. A theory beyond the Standard Model should be able to explain this. All these arguments tend to indicate that the Standard Model may be an approximation describing low-energy aspects of a more fundamental theory. The difference between the approximation and the absolute theory would then start to appear when the energies involved become large.

The most popular extension is called super-symmetry (SUSY). It predicts the existence of massive super-symmetric particles, so-called *s* particles, for each fundamental particle. There is lack of experimental evidence of these particles. Another one is the string theory. Behind this theory is the idea that fundamental particles are small loops of vibrating strings. All the different particles and forces are just different oscillation modes of a unique type of string. The theory also implies that besides the familiar four dimensions, there are six additional spatial dimensions.

1.2. CERN

CERN [2] is the European Organization for Nuclear Research and it is the world’s largest particle physics centre. The word CERN stands for *Conseil Européen pour la Reserche Nucléaire*, a provisional body founded in 1952 for establishing world-class fundamental physics research organization in Europe. The Organization was officially founded in 1954 and the Council was dissolved. The new organization was given the title European Organization for Nuclear Research,

although the name CERN was retained. Currently, CERN is composed by 20 European Member States, but many non-European countries are also involved in different ways. It was one of Europe's first joint ventures and has become a shining example of international collaboration. The laboratory is located in the Franco-Swiss border close to Geneva. The CERN laboratory provides state-of-the-art scientific facilities for researching into the basic building blocks of the Universe. These are for instance accelerators, which accelerate tiny particles to a fraction under the speed of light, and detectors to make the particles visible.

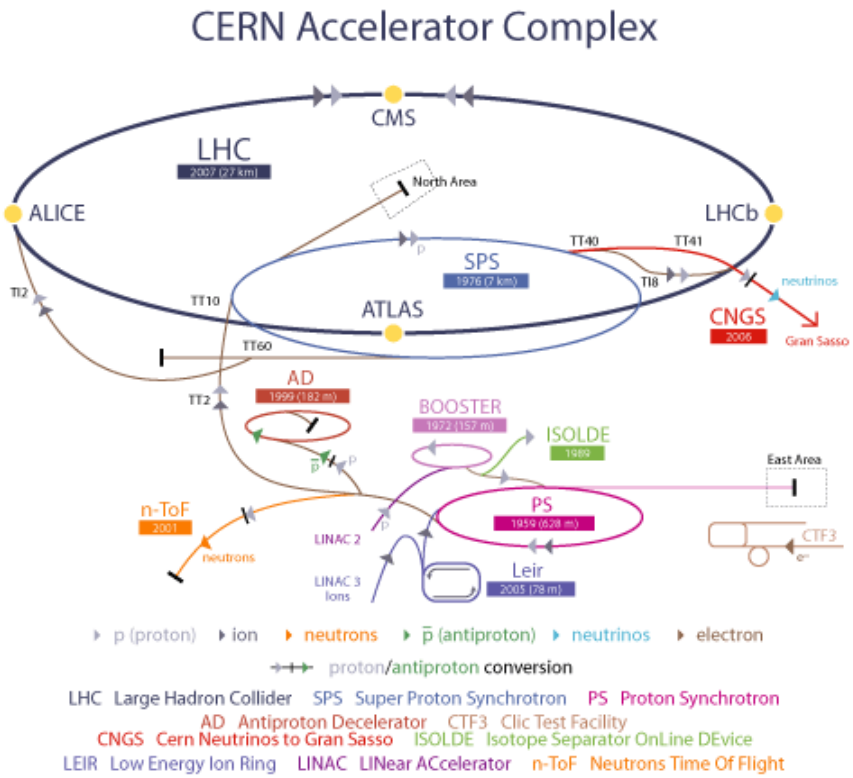

Figure 1.1. *CERN accelerator complex.*

The Proton Synchrotron (PS), which was the first accelerator, began operation in 1959 and continues in operation, at the heart of CERN accelerator complex. This complex (see figure 1.1) is composed by a group of accelerators linked consecutively. On each step, a new accelerator boosts the speed of a beam of particles, before injecting it into the next one in the sequence, increasing the energy of the particles in order to achieve the required energy in the last step. The linear accelerators, so-called LINACs, form the first step. Then circular accelerators are

used since they can accelerate the particles to higher energies in a relatively little space when compared to linear accelerators. The CERN's most important circular accelerators are the Proton Synchroton (PS), the Booster (also known as PS Booster), the Super Proton Synchrotron (SPS) and, of course, the Large Hadron Collider (LHC), the newest and most powerful accelerator.

The complex includes smaller machines, particularly the Antiproton Decelerator (AD) and the On-Line Isotope Mass Separator (ISOLDE) facility. These allow a wide range of investigations into exotic forms of matter and anti-matter. The PS and SPS also feed the CERN Neutrinos to Gran Sasso (CNGS) project, which sends neutrinos underground all the way to Italy. Protons are obtained by removing electrons from hydrogen atoms. They are injected from the linear accelerator LINAC2 into the PS Booster, then the PS, followed by the SPS, before finally reaching the LHC. On the other hand, lead ions for the LHC start from a source of vaporized lead and enter LINAC3 before being collected and accelerated in the Low Energy Ion Ring (LEIR). They follow then the same route to maximum acceleration as the protons.

1.3. The Large Hadron Collider (LHC)

The Large Hadron Collider (LHC) [3, 4] is an accelerator which brings either protons or lead ions into head-on collisions at higher energies than ever achieved before. For instance, in the case of proton-proton collisions, two beams of protons will each be accelerated at a maximum energy of 7 TeV, corresponding to head-to-head collisions of 14 TeV, that is, seven times higher than Tevatron, located in Fermilab (USA). In case of lead ions collisions, the collision energy will reach 1150 TeV. This will allow exploring the physics at the TeV scale. The CERN's accelerators complex provides the accelerated particles that the LHC uses, as shown in figure 1.1. The LHC has a length of 27 km that corresponds to a radius of 4.3 km. It is located 100 m underground, taking advantage of the previous Large Electron-Positron Collider (LEP) tunnel.

The LHC started up in September of 2008 when the first beam of protons was steered around the accelerator. However, the first collisions did not take place until November of 2009, after a stop of one year. During 2010, the LHC worked successfully, increasing gradually the energy of the proton-proton collisions up to 7 TeV, reaching the targeted peak luminosity of $2 \cdot 10^{32}$ $cm^{-2}s^{-1}$ for proton-proton collisions and starting the lead-ion collisions. During 2011, a peak luminosity of $3.6 \cdot 10^{33}$ $cm^{-2}s^{-1}$ for proton-proton collisions has been reached.

1.3.1. Main characteristics of the accelerator

The proton beams are accelerated using radio-frequency (RF) cavities. This type of cavities accelerates the charged particles with a time-varying electrical field. The particles have to be injected into the cavity at the correct phase, producing bunched beams instead of continuous beam, since it is not possible to accelerate a continuous beam. The particles are maintained in a circular path by means of dipole magnets. The momentum of the particles in a circular accelerator can be expressed as:

$$p = 0.3 \cdot B \cdot \rho \quad (1.2)$$

where B is the magnetic field in Tesla, ρ is the radius of curvature in m and p is the momentum in GeV/c. High energy LHC beams need high magnetic bending fields, because the machine radius was a fixed parameter determined by the tunnel. To bend 7 TeV protons around the ring, the LHC dipoles must be able to produce fields of 8.33 T. Thus, superconducting magnets are used in order to provide such high magnetic field.

Another important parameter of the LHC is the luminosity, L. It denotes the frequency of collisions per unit of cross-section at the interaction point. Therefore it is related to the reaction rate of particles, R, created in a certain process. This can be expressed as:

$$R = \frac{dN}{dt} = \sigma \cdot L \quad (1.3)$$

where σ is the cross-section of that process. The luminosity depends on the number of particles per bunch in the colliding beams, N_1 and N_2, on the bunch crossing frequency f and on the bunch area:

$$L = \frac{N_1 \cdot N_2 \cdot k_b}{4 \cdot \pi \cdot \sigma_x \cdot \sigma_y} \cdot f \quad (1.4)$$

where k_b is the number of bunches and σ_x and σ_y are respectively the Gaussian sizes of the beam in the transverse plane. The designed luminosity of the LHC is $L = 10^{34}$ cm^{-2}s^{-1}. This luminosity is achieved with a bunch intensity of 1.15 x 10^{11} protons and 2808 bunches per beam. When colliding lead ions, the luminosity reduces to $L = 10^{27}$ cm^{-2}s^{-1}. In order to achieve this level of luminosity the LHC works at a frequency of 40 MHz in case of proton-proton collisions, that is, the bunches collide every 25 ns.

The LHC has eight arc sections and eight straight sections with experiments and systems for machine operation, as shown in figure 1.2. The two counter-rotating beams collide in the centre of the experimental detectors in four of the straight sections with a crossing angle of 300 μrad. There are insertions along the tunnel for the experiments, for machine operation, beam cleaning, beam dumping, RF acceleration and beam instrumentation.

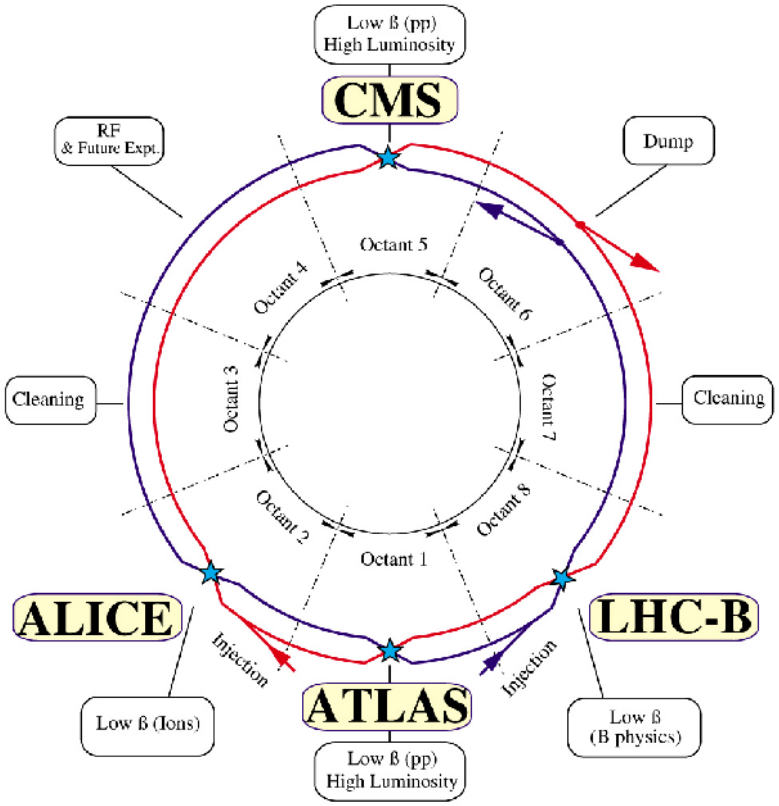

Figure 1.2. *LHC schematic layout.*

Two counter-rotating proton or lead ion beams are collided in the LHC. These beams need opposite deflecting magnetic fields in the arcs. The technology used is superconducting twin aperture magnets (see figure 1.3). These are the magnets that produce the required magnetic field of 8.33 T with a current of 11850 A. The two magnetic channels are housed in the same yoke and cryostat, a unique configuration that saves both space and cost over separate rings. The beam pipes separation is 194 mm. The magnet is curved in the horizontal plane with a bending

angle of 5.1 mrad that corresponds to a radius of curvature of about 2812 m at 293 K, in order to closely match the trajectory of the particles. The total length of the magnet is 14.3 m. There are 1232 main dipole magnets filling more than 2/3 of the LHC ring.

The coils of the magnet are made of copper-clad niobium-titanium cables. Using this technology, a superconducting regime, characterized by a minimal electrical resistance, can be reached in the wires. In order to keep the cable on superconducting state, three important parameters must be controlled. These parameters are the temperature, the magnetic field and the current density. For instance, the operating temperature required for the magnets is about 1.9 K. This temperature value requires a cooling system using super-fluid helium, which has very low viscosity and very high heat conduction.

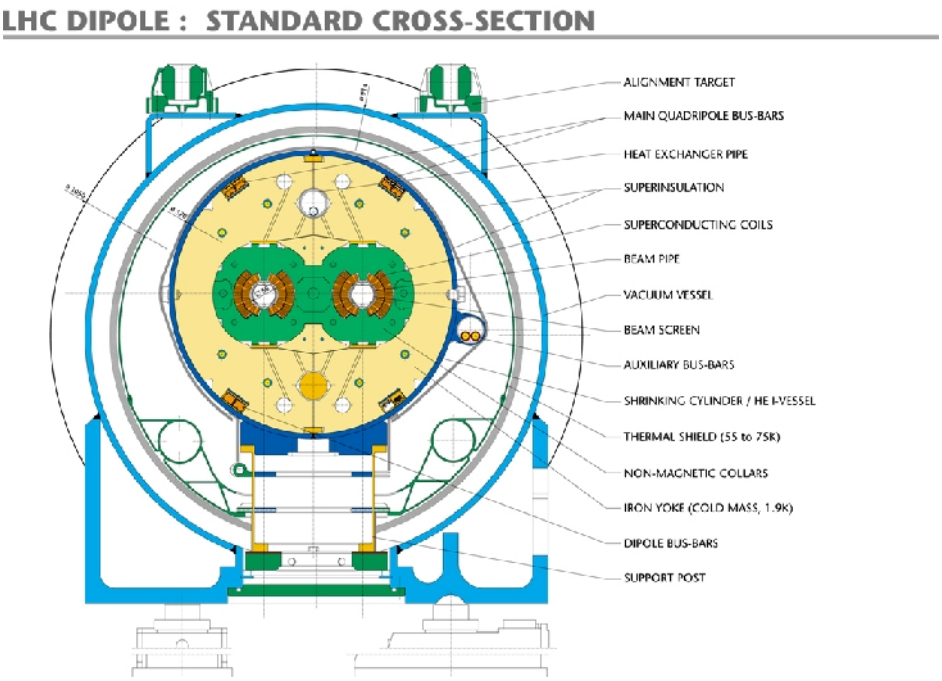

Figure 1.3. *Cross-section of a LHC cryodipole.*

As well as the main dipole magnets, other superconducting multipolar magnets are needed for beam focusing and correction of field deviations. In order to focus the beam, 392 quadrupole magnets are used. These quadrupole magnets have the same cable type for the winding as the dipole magnets. Each of the eight LHC arcs contain 23 regular cells, having each cell six dipole magnets and two quadrupole

magnets with opposite polarities for focusing the particles in both planes. Finally, beam stability is achieved by means of corrector magnets. These corrector magnets are sextupole, octupole and decapole magnets. Table 1.4 summarizes some of the LHC collider parameters discussed.

1.3.2. LHC experiments

Six experiments operate at the LHC. The largest experiments are ATLAS (A large Toroidal LHC ApparatuS) [5] and CMS (The Compact Muon Solenoid) [6]. There are also two medium size experiments like ALICE (A Large Ion Collider Experiment) [7] and LHCb (Large Hadron Collider beauty experiment) [8]. Finally, three smaller experiments as TOTEM (Total Cross Section, Elastic Scattering and Diffraction Dissociation at the LHC) [9], LHCf (Large Hadron Collider forward) [10] and ALPHA anti-hydrogen experiment. The ATLAS, CMS, ALICE and LHCb detectors are installed in four underground caverns located around the tunnel of the LHC (see figure 1.2). The detectors used by the TOTEM experiment are positioned near the CMS detector, whereas those used by LHCf are near the ATLAS detector.

ATLAS is a general purpose detector. It will explore a wide range of physics including the search of the Higgs boson, extra dimensions and particles that could compose dark matter. The detector optimization is guided by physics issues such as sensitivity to the largest possible Higgs mass range. Other important goals are the searches of heavy bosons (W' and Z'), of super-symmetric particles, of compositeness of the fundamental fermions, the investigation of *CP* violation in *B*-decays as well as detailed studies of the top quark.

One of the main features of ATLAS is its enormous magnet system. This consists of eight 26 m long superconducting magnet coils, arranged to form a cylinder around the beam pipe through the centre of the detector. During operation, the magnetic field is contained within the central cylindrical space defined by the coils. Regarding the ATLAS detectors system, it comprises three main sub-systems, with different detecting technologies within each of them. From the inside-out:

- the *Inner Detector*, pattern recognition, momentum and vertex measurements as well as enhanced electron identification are achieved with a combination of discrete high-resolution pixel and microstrip silicon detectors in the inner part and continuous straw-tube tracking detectors in the outer part of the tracking volume,

- the *Calorimetry System*, an electromagnetic calorimeter for the identification and energy measurement of electrons and photons, and a

hadronic calorimeter for the measurement of hadronic jets and missing energy,

- the *Muon Spectrometer*, a stand-alone tracking device for muon detection including different high-precision tracking chambers for an excellent measurement of the muon momenta and trigger chambers with very fast response.

Parameter	**p ↔ p**	**Pb ↔ Pb**
Collision energy	7 TeV	2.76 TeV/n
Centre-of-mass collision energy	14 TeV	1150 TeV
Injection energy	450 GeV	1.774 TeV/n
Luminosity	10^{34} cm^{-2}s^{-1}	10^{27} cm^{-2}s^{-1}
Operation frequency	40 MHz	10 MHz
Bunch separation	25 ns	100 ns
Number of bunches	2808	592
Number of particles per bunch	1.15×10^{11}	7×10^{7}
Beam current	0.56 A	
Circumference	26.658883 m	
Radius	4.242893 m	
Number of dipole magnets	1232	
Dipole magnet length	14.3 m	
Magnetic field	8.33 T	
Total mass	27.5 t	

Table 1.4. *Main nominal parameters of the LHC collider.*

The CMS experiment is also a general purpose detector, like ATLAS. It has the same scientific goals as the ATLAS experiment but it uses different technical solutions and design of its detector magnet system to achieve these. The CMS detector is built around a huge solenoid magnet. This takes the form of a cylindrical coil of superconducting cable that generates a magnetic field of 4 T. The magnetic field is confined by a steel yoke. The detector system of CMS also has the same sub-systems as ATLAS, an all silicon *Inner Tracking Detector* (with pixel and microstrip silicon detectors), a *Calorimetry System* (with an electromagnetic calorimeter and a hadronic calorimeter) and a *Muon Detector System*.

The ALICE experiment has been developed for the collision of lead ions. The main goal is recreating the conditions just after the Big Bang under laboratory

conditions. It is expected that under these conditions, the protons and neutrons will *melt*, freeing the quarks from their bonds with the gluons. This should create a state of matter called quark-gluon plasma, which probably existed just after the Big Bang when the Universe was still extremely hot. The ALICE collaboration plans to study the quark-gluon plasma as it expands and cools in order to observe how it progressively gives rise to the particles that constitute the matter of the Universe. In the ALICE detector most of the detection elements are partially or entirely enclosed in a huge solenoid magnet whose weak magnetic field of 0.5 T bends the path of the charged particles generated in collisions, allowing for the determination of their electric sign and momentum. The detection system of the ALICE experiment is composed of different sub-systems like tracking systems (as the *Inner Tracking System* which uses microstrip silicon detectors), transition radiation detectors, time of flight detectors, photon and muon spectrometers, a photon multiplicity detector or the trigger system.

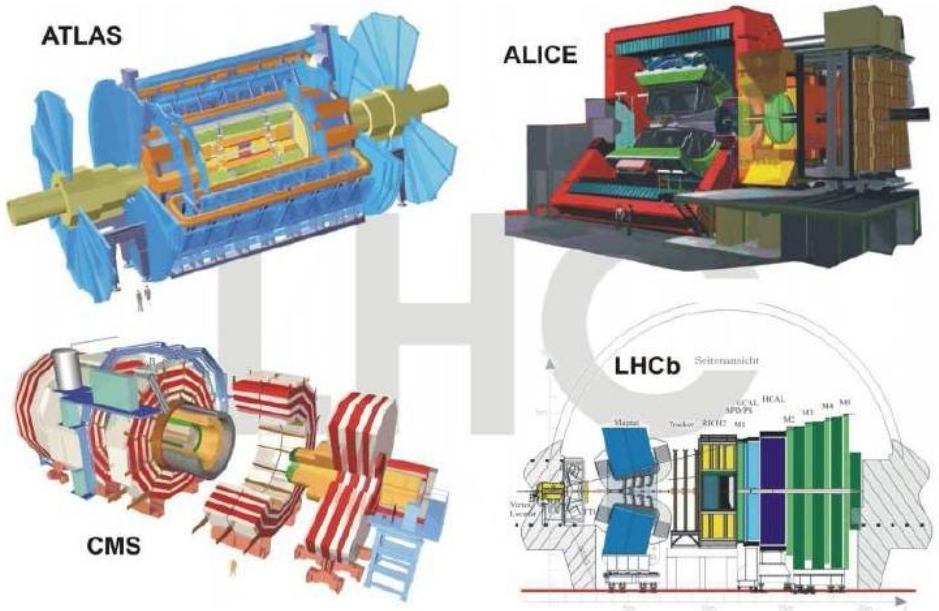

Figure 1.4. *Three dimensional views of ATLAS, ALICE and CMS experiments. Longitudinal cross-section of LHCb experiment.*

The LHCb experiment is designed to exploit the large number of *b* quarks produced at LHC in order to make precision studies of *CP* asymmetries and of rare decays in the *B*-meson systems. Currently, observed *CP* violation in neutral kaon decays can be accommodated within the Standard Model. However, it cannot be excluded that physics beyond the standard model contributes or even fully accounts

for the *CP* violation. Furthermore, *CP* violation also plays an important role in cosmology. It is one of the ingredients required to explain the excess of matter over antimatter observed in our universe.

LHCb is a single-arm spectrometer. Instead of surrounding the entire collision point with an enclosed detector, the LHCb experiment uses a series of sub-detectors to detect mainly forward particles. The first sub-detector is mounted close to the collision point, while the next ones stand one behind the other, over a length of 20 m. The spectrometer dipole is placed close to the interaction region, in order to keep its size small. A superconducting magnet is used to produce a vertical magnetic field of 1.1 T. The polarity of this field can be changed. Regarding the detection system, LHCb comprises the different sub-detectors as the *Vertex Detector System* (which uses microstrip silicon detectors), the *Tracking System* (also uses microstrip silicon detectors as well as straw-tubes) , the *RICH Detectors*, the *Calorimetry System* or the *Muon Detector.*

TOTEM is an experiment dedicated to the measurement of total cross section, elastic scattering and diffractive processes at the LHC. The total cross section will be measured using the luminosity independent method which is based on the simultaneous detection of elastic scattering at low momentum transfer and of the inelastic interactions. This method also provides an absolute calibration of the machine luminosity.

Finally, the LHCf experiment uses forward particles created inside the LHC as a source to simulate cosmic rays in laboratory conditions. Cosmic rays are charged particles from outer space that constantly interact with the atmosphere. These particles collide with nuclei in the atmosphere, leading to a cascade of particles that reaches ground level. Studying how collisions inside the LHC cause similar cascades of particles will help to interpret and calibrate large-scale cosmic-ray experiments that can cover thousands of kilometres.

1.4. The High Luminosity LHC

The LHC experiments started their operation in 2009. As stated above, the LHC detectors have been designed for a luminosity of 10^{34} $cm^{-2}s^{-1}$. If the Higgs has been observed then as additional luminosity accumulates, parameters such as the mass and couplings will need to be measured [11]. There will be insufficient statistics to measure the Higgs self-coupling at the LHC. This may require more luminosity. Furthermore, if SUSY is observed at the LHC, then the masses and model will need to be determined, along with the connection to cosmology, the impact on Higgs phenomenology and the SUSY breaking mechanism. If neither the Higgs nor

SUSY is observed, then other possibilities would need investigation (extra dimensions, little Higgs models, quarks structure, etc). Much of this is difficult with the designed LHC.

Some studies about an upgrade of the LHC have shown that an increase of the luminosity by a factor of ten would be possible. This upgrade would improve the physics potential for the LHC experiments and it is known as High Luminosity LHC (HL-LHC).

The main characteristic of the HL-LHC is an increase of the luminosity from 10^{34} $cm^{-2}s^{-1}$ to 10^{35} $cm^{-2}s^{-1}$. The number of bunches per beam would be doubled. The expected integrated luminosity for HL-LHC would be about 3000 fb^{-1} per experiment.

The LHC upgrade can be classified in three phases [12].

- **Phase 0:** the LHC would be taken to its maximum performance without hardware changes, yielding a luminosity of $2.3 \cdot 10^{34}$ $cm^{-2}s^{-1}$. The LHC dipole field could be increased to a value of 9 T, raising the energy from 7 to 7.54 TeV.

- **Phase 1:** the insertion quadrupoles would be modified, the number of protons per bunch would be increased to its maximum value and the number of bunches would be doubled. This could yield a luminosity of $9.2 \cdot 10^{34}$ $cm^{-2}s^{-1}$.

- **Phase 2:** energy and luminosity upgrade involving injection systems modifications, SPS with superconducting magnets and upgrading the transfer lines in order to produce an energy of 14 TeV. The luminosity would reach its maximum value of 10^{35} $cm^{-2}s^{-1}$.

The HL-LHC presents a number of experimental challenges such as radiation damage to detectors and an increase in pile-up of additional overlapping events. The ATLAS and CMS tracking systems would need replacements, both detectors and electronic systems. The silicon detectors will suffer important radiation damage, so they will need to be completely replaced. The calorimeters will need modifications and new electronic systems as well. The muon systems may require some improvements. Finally, ATLAS and CMS trigger and DAQ systems would need significant modifications to operate at the HL-LHC.

1.5. The RD50 collaboration and the ALIBAVA collaboration

Silicon detectors are devices used for accurate track reconstruction of charged particles. As has been shown before, the requirements at the LHC have pushed the present day silicon tracking detectors to the very edge of the current technology. Future very high luminosity colliders or a possible upgrade scenario of the LHC to a luminosity of 10^{35} $cm^{-2}s^{-1}$ will require semiconductor detectors with substantially improved properties. Considering the expected total fluences of fast hadrons above 10^{16} cm^{-2} and an increase in the number of bunches per beam, the detector must be ultra radiation hard, provide a fast and efficient charge collection and be as thin as possible.

The CERN RD50 Collaboration [13] is currently addressing this need. It is a collaboration for research and development to provide a detector technology, which can operate safely and efficiently in an environment as described above. The main aims are to optimize existing methods and evaluate new ways in order to engineer the silicon bulk material, the detector structure and the detector operational conditions. Also, possibilities to use semiconductor materials other than silicon are being explored.

The RD50 collaboration was created in 2001 and approved in 2002. Presently, RD50 counts a total of 253 members with 48 participating institutes. This comprises 39 institutes from 18 different countries in West and East Europe, 8 from North America (USA, Canada) and one from Middle East (Israel). The RD50 collaboration is the continuation of the ROSE Collaboration (Research and development On Silicon for future Experiments). The ROSE collaboration was started in 1996 and it concluded in 2000. The objectives of the collaboration were the development of radiation hard silicon detectors that could operate beyond the limits of the period devices and that ensured guaranteed operation for the whole lifetime of the LHC experimental programme, and also the outline of recommendations to experiments on the optimum silicon for detectors and quality control procedures required to ensure optimal radiation tolerance.

The RD50 collaboration has two main research lines: *Material Engineering* and *Device Engineering*. Each of these two main lines is divided in three projects as can be seen in figure 1.5. Material engineering stands for the deliberate modification of the detector bulk material, either by means of the silicon defect engineering (which for example includes the enrichment of the silicon base material with oxygen, oxygen dimers or other impurities) or the use of other

semiconductor materials than Silicon (SiC or GaN). Device engineering stands for the design of devices for future experiments. The characterization of pad detectors, microstrip and pixel detectors in a hard radiation environment is a main objective as well as the development of new detector geometries, as 3D detectors or thin detectors.

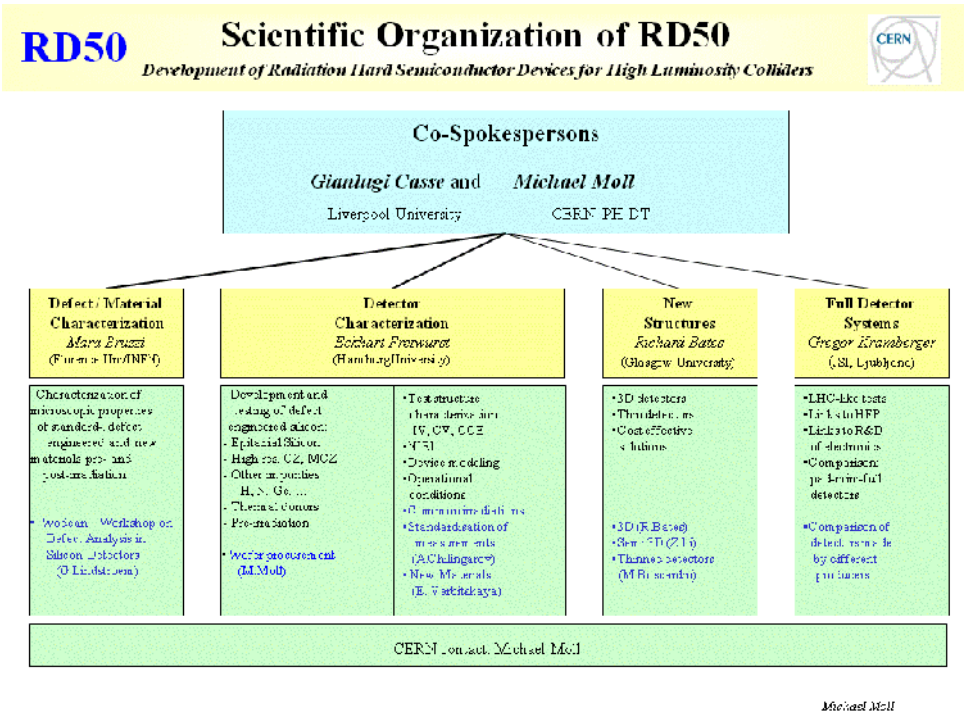

Figure 1.5. *Main research lines of the RD50 Collaboration.*

The Full Detector Systems project can be considered as the main frame for the development of this thesis. Among other lines, this line is devoted to the following issues.

- Taking advantage of the LHC speed electronics to test the ultimate parameters for the present segmented detectors. Study of the lifetime limit of the system as a whole. Study the possible improvement of the present system to be implemented.

- Systematic evaluation of segmented detectors made of new RD50 materials and study of new structures.

- Anticipation of the new problems connected with HL-LHC. Links to R&D of electronics.

- Device simulation of segmented devices.

- Design and realization of radiation hard pixel and microstrip detectors.

- Setup of low noise systems to measure microstrip and pixels made with thin detectors or new material to characterized very low signals.

- Comparison of pad-mini-full detectors.

Under the *Full Detector Systems* research line a collaboration among the *University of Liverpool*, the *Centro Nacional de Microelectrónica (CNM)* of Barcelona and the *Instituto de Física Corpuscular* (IFIC) of Valencia was created in 2005. This collaboration is so-called ALIBAVA. The first aim of this collaboration was the design, development and implementation of a readout system for microstrip silicon sensors using an electronic system similar to those used in the LHC experiments. In the next chapters the design, development and implementation of this system is explained in detail as well as its upgrade for taking measurements in a test beam.

Chapter 2

Silicon Sensors

An overview of the principles of silicon detectors is given in this chapter, since the system whose design is detailed in this thesis uses this type of sensors. The main properties of the silicon as reference semiconductor material are explained in section 2.1. In section 2.2, the pn junction behaviour is detailed as it remains as the basic layout for this kind of detectors. The pn junction characteristics of interest for silicon detectors, as depletion voltage, capacitance or leakage current are described. Section 2.3 is devoted to the principles of operation of this type of sensors, treating of the energy loss of charged particles traversing a solid, the signal collection process and the position resolution. A brief summary of the radiation damage effects is carried out in section 2.4. The structures used for producing silicon detectors are presented in section 2.5. Subsequently, the front-end electronics stages used for the signal acquisition are described in section 2.6. In particular, the amplification and the pulse shaping are explained. Finally, the noise sources of the silicon sensors and the front-end electronics are explained and quantified in section 2.7.

2.1. Physical properties of silicon

Semiconductor materials, like silicon or germanium, are based on solid crystals where the atoms in the lattice create covalent bonds with its four neighbouring atoms. In particular, silicon atoms are arranged in a tetrahedral configuration in a diamond cubic lattice. Silicon wafers used for detectors can be cut oriented at a specific crystallographic direction since the properties of the crystal vary along different planes. Common orientations are parallel to the face of the cubic lattice or

following a diagonal plane defined by three non-adjacent corners of the lattice [14].

The conductivity of a semiconductor can be explained in terms of the band theory of solids. Following the quantum theory, the electrons in the atoms have discrete energy levels perfectly defined when they are widely separated. However, when a solid crystal is considered, in which the inter-atomic spacing is small, the discrete levels allowed for the electrons become energy bands (figure 2.1). These energy bands are composed by many discrete levels slightly separated from each other because of the interactions among the outer layers of electrons of each atom.

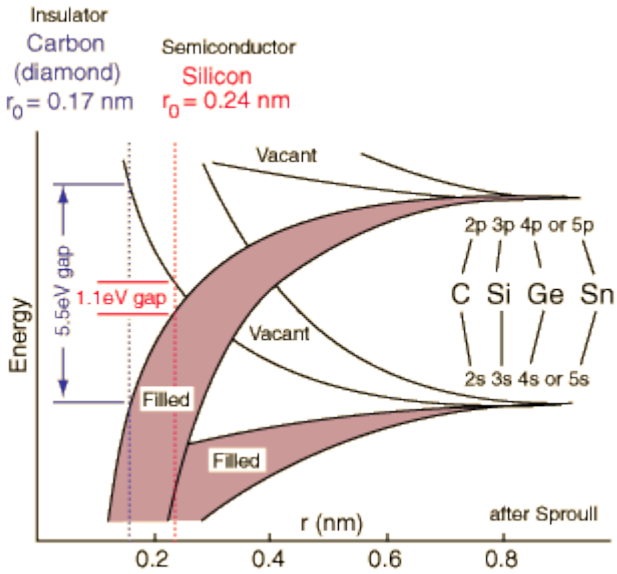

Figure 2.1. *Energy band diagram for silicon and carbon as a function of the inter-atomic distance. As the inter-atomic distance decreases, the bonding states (filled orbitals) and anti-bonding states (empty orbitals) spread to form the valence band, a forbidden gap and a conduction band. Taken from [15].*

The allowed bands for the electrons are separated by energy gaps. The highest energy band is the conduction band. This band contains free electrons contributing to the electrical conductivity of the material. However, the valence band contains tightly bound electrons. The energy gap between the conduction band and the valence band, the forbidden gap, depends on the type of material. In insulators, the electrons in the valence band are separated by a large gap (3.5-6 eV) from the conduction band while in conductors like metals, the valence band overlaps the conduction band. In semiconductors, there is a small enough gap (~1 eV) between the valence and conduction bands that thermal or other excitations can bridge the

gap.

For silicon at 0 K, all energy states in the valence band are filled by electrons which participate in covalent bonds among the lattice atoms, so the conduction band is empty. At higher temperatures, a number of electrons that are thermally excited can reach the conduction band, leaving a vacancy known as a hole, in their original positions. These holes behave as positive charges. Another electron from a neighbouring atom can fill the hole, creating then a new hole in the process. Therefore, there will be a current of electrons in the conduction band and a current of holes in the valence band, contributing both charge carriers to the electrical conductivity of the silicon.

On the contrary, there is a process known as recombination whereby electrons from the conduction band occupy, in one or multiple steps, the empty state associated with a hole in the valence band releasing energy in the process. The concentration of electrons (n) and holes (p) is the same at thermal equilibrium, so the intrinsic concentration of charge carriers (n_i) is used instead. This concentration is strongly dependent on the temperature and it is given by

$$n_i^2 = n \cdot p = N_c \cdot N_v \cdot \mathrm{e}^{\left(-\frac{E_g}{k \cdot T}\right)} \quad (2.1)$$

where N_C and N_V are the effective density of states in the conduction band and the valence band respectively, k is the Boltzmann constant, T is the absolute temperature and E_g is the energy gap (E_g for silicon is 1.12 eV and for germanium is 0.7 eV). The pure silicon, as pure germanium, is an intrinsic semiconductor. The pure silicon intrinsic carrier density n_i has a value of 1.45 x 10^{10} cm^3 at room temperature (300 K) [14]. This low concentration implies low conductivity.

Some impurity atoms or dopants are usually added in order to modify the electrical properties of the material, for instance, increasing the conductivity of a semiconductor. Atoms with valence of three or five are introduced into the silicon lattice in a process known as doping. As a result, an extrinsic semiconductor is obtained, which is doped. If silicon is doped with pentavalent atoms (P, As, Sb), it is called n-type because of the excess of electrons introduced by the dopant (donor atoms). On the other hand, a p-type semiconductor is doped with trivalent impurities (B, Al, Ga) becoming holes the majority charge carriers, since trivalent atoms are acceptor elements. The energetic levels introduced by the dopant atoms are called shallow levels, since they are situated very close to the conduction band or valence band depending on the dopant type (figure 2.2).

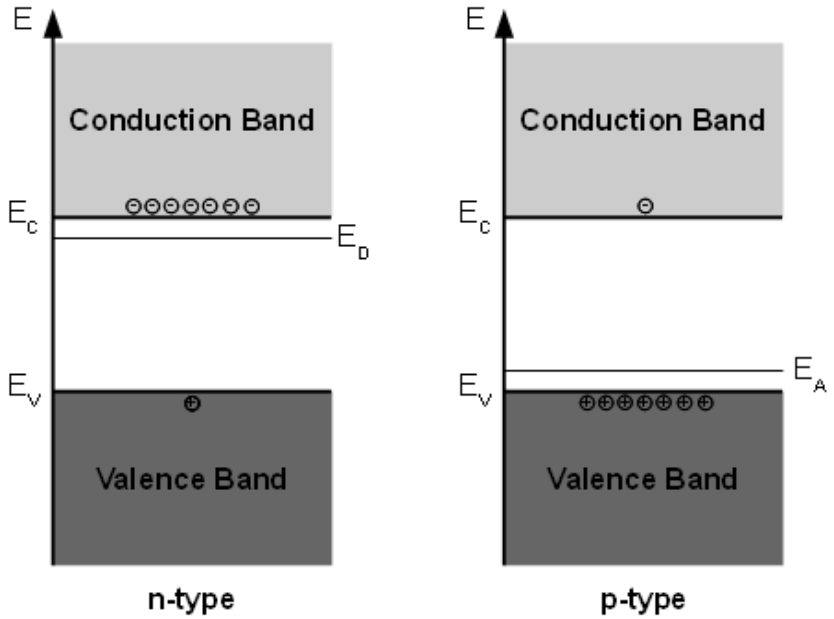

Figure 2.2. *Energy band diagram for n-type (left) and p-type (right) semiconductors. In the n-type, the donor levels (E_D) are close to the conduction band edge (E_C), so thermal excitation can promote electrons into the conduction band. In the p-type, the acceptor levels (E_A) lie just above valence band edge (E_V), so thermal excitation can promote electrons from the valence band to fill the acceptor state, leaving holes in the valence band.*

The carrier type introduced by the impurities dominates the conductivity whether the doping concentration is larger than the intrinsic carrier concentration, as suggests the following equation:

$$N_a + n = N_d + p\,. \quad (2.2)$$

Here N_a denotes the acceptor atoms concentration and N_d the donor atoms concentration. Silicon and other semiconductor crystals suffer from imperfections during crystal growth, during device fabrication or by radiation damage. Defects in the crystal such as impurity atoms, vacancies and structural irregularities can create energy states throughout the band gap. Those states closer to the middle of the band gap can act as generation and recombination centres, contributing to the emission and capture of charge carriers.

2.2. The *pn* junction

The main structure used in silicon sensors is basically a diode operated under reverse bias. The *pn* junction consists of two semiconductor materials with different dopant concentrations joined together. This junction is usually formed by combining a *p*-type semiconductor material with a *n*-type material in very close contact.

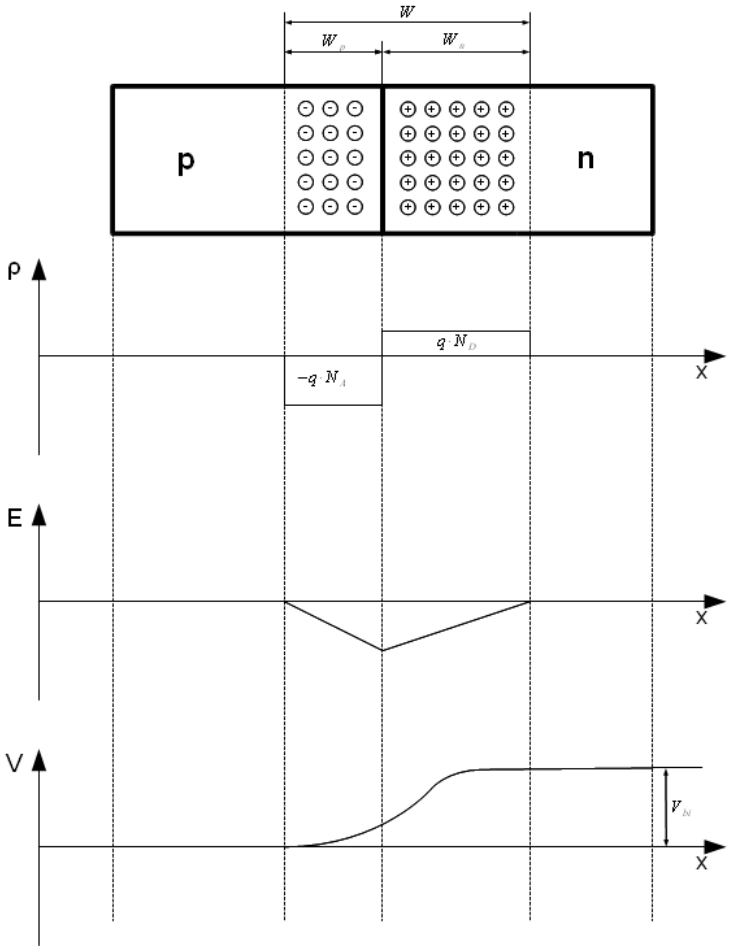

Figure 2.3. *A pn junction in thermal equilibrium. An abrupt change between a neutral and a completely depleted space charge zone is assumed Under the junction, plots for the charge density, the electric field and the voltage are reported.*

Because of the strong gradient in the concentration of electrons and holes in both

sides of the junction, there will be a diffusion of charge carriers across the junction (electrons to the *p*-type region and holes to the *n*-type region). Since the materials forming the junction are electrically neutral overall, the recombination of charge carriers crossing the junction leaves fixed ions of opposite charge on either sides of the junction (positive ions in the *n* region and negative ions in the *p* region). An electric field is then created which will stop the diffusion and will leave a region in between free of charge carriers. This region is so-called the depletion zone or space charge region.

Therefore, in a *pn* junction without an external applied voltage, an equilibrium condition is reached in which a potential difference is formed across the junction (figure 2.3). This potential difference is called built-in potential, V_{bi}, it depends logarithmically on doping level [14, 16]

$$V_{bi} = \frac{k \cdot T}{q} \cdot \ln\left(\frac{N_a \cdot N_d}{n_i^2}\right) \quad (2.3)$$

where q is the electron charge. The built-in potential is about 0.6 V for silicon.

2.2.1. Depletion width

An ionizing particle traversing the depleted region of a *pn* junction diode will release free carriers (electron-hole pairs). The depletion zone is a volume without mobile carriers and it forms a capacitor, where the undepleted *p* and *n* regions act as electrodes and the depletion region is the dielectric. An electric field in this region will sweep the generated free carriers to the electrodes, so the diode forms an ionization chamber. At thermal equilibrium, the low built-in potential of the *pn* junction is not able to produce a large enough electric field to generate a quick net movement of the charge carriers. Thus, charges can be lost due to trapping and recombination, resulting in an incomplete charge collection. Nevertheless, both the magnitude of the electric field and the width of the depletion region can be increased, if an external reverse potential difference is applied to the *pn* junction. As a result, there will be a more efficient charge collection and an increase of the detection sensitive volume.

Assuming an abrupt junction, where the charge densities are uniform in the depletion zone ($N_d \cdot q$ in the *n* side and $N_a \cdot q$ in the *p* side as shown in figure 2.3), the width of the depletion region in both the *n* side, W_n, and the *p* side, W_p, of the junction can be derived from the solution to the Poisson's equation [16, 17]. W_n is given by

$$W_n = \sqrt{\frac{2 \cdot \varepsilon_{Si} \cdot (V_{bi} + V_{bias}) \cdot N_a}{q \cdot N_d \cdot (N_a + N_d)}} \quad (2.4)$$

and *Wp* is given by

$$W_p = \sqrt{\frac{2 \cdot \varepsilon_{Si} \cdot (V_{bi} + V_{bias}) \cdot N_d}{q \cdot N_a \cdot (N_a + N_d)}} \quad (2.5)$$

with V_{bias} the reverse voltage applied to the junction and ε_{Si} the silicon permittivity. The total depletion width is then:

$$W = W_n + W_p = \sqrt{\frac{2 \cdot \varepsilon_{Si} \cdot (V_{bi} + V_{bias}) \cdot (N_d + N_a)}{q \cdot N_d \cdot N_a}}. \quad (2.6)$$

The width of the depletion zone in each of the two regions (*p* region and *n* region) is inversely proportional to the concentration of impurities. Therefore, the extent of the depletion zone can be controlled by choosing different doping concentrations on both sides of the junction. The more highly doped is one side, the more extent is the depletion zone on the other. Detector diodes are usually doped asymmetrically. For instance, if the *p* side is highly doped, $Na \gg Nd$, the depletion zone would extend almost entirely over the lightly doped *n*-side and *W* reduces to

$$W \simeq \sqrt{\frac{2 \cdot \varepsilon_{Si} \cdot (V_{bi} + V_{bias})}{q \cdot N_d}}. \quad (2.7)$$

This is the case of a p^+-*n* junction. The width of the depleted region in the *n* side of the junction can be increased by reducing the dopant concentration. This is limited by a minimum level of impurities in the crystal. Therefore, for the *n* side of the p^+-*n* junction a effective charge concentration or net dopant concentration can be defined as

$$N_{eff} = |N_d - N_{a,n}| \quad (2.8)$$

where $N_{a,n}$ is the acceptor concentration in the *n*-side of the p^+-*n* junction.

The abrupt junction approximation assumes that the impurity concentration changes sharply from the *p*-side to the *n*-side. Nevertheless, in practice, the change

between the two regions occurs gradually over a finite distance and the *Debye-tails* of the carrier distribution at the edges of the depleted region should be taken into account. The abrupt junction approximation is adequate as long as these transition regions are thin compared to the depletion region width [14, 16], which is the case of silicon detectors.

2.2.2. Capacitance and full depletion voltage

As it has been mentioned previously, a reversed *pn* junction has properties of a charged capacitor as a result of the depleted region free of mobile charges bounded by conducting *p*-type and *n*-type semiconductors on each side. An increase of the reverse voltage (*dV)* enlarges the depletion zone, resulting in a charge increment (*dQ)* on either sides. Considering a reversed biased p^+-*n* junction as a parallel plate capacitor, the junction capacitance can be calculated using equation 2.7 and 2.8 as

$$C = \frac{dQ}{dV} = \varepsilon_{Si} \cdot \left(\frac{A}{W} \right) = A \cdot \sqrt{\frac{q \cdot \varepsilon_{Si} \cdot N_{eff}}{(V_{bi} + V_{bias})}} \quad (2.9)$$

where *A* is the junction surface. The capacitance decreases proportionally to the reverse voltage applied to the junction until the depletion region extends to the full width of the junction. The depletion region can be increased to the total width of the junction, *d.* The reverse bias voltage where full depletion is reached is known as the full depletion voltage, V_{FD}, and can be calculated as

$$V_{FD} = \frac{q \cdot d^2 \cdot N_{eff}}{2 \cdot \varepsilon_{Si}} \quad (2.10)$$

with *d* the total width of the detector. For a reverse voltage higher than V_{FD}, the capacitance remains constant and corresponds to the geometrical capacitance of a parallel plate capacitor with spacing equivalent to the total width of the junction

$$C = \varepsilon_{Si} \cdot \left(\frac{A}{d} \right). \quad (2.11)$$

2.2.3. Leakage current and breakdown voltage

Although ideally, a reverse bias voltage applied to enlarge the depletion zone would remove all the mobile carriers from the *pn* junction volume, there will be small current flows even without the passage of a charged particle. This leakage or

reverse current is the basis of the background noise of silicon detectors. The leakage current has two contributions, the thermal generation current and the diffusion current.

The main source of this current is thermally generated electron-hole pairs within the depleted region. The probability of electrons transitioning from the valence band to the conduction band is increased by the presence of impurities in the silicon lattice since they introduce intermediate energy states in the gap. The electric field in the depletion region prevents charge carriers to recombine since they are swept to the electrodes, increasing the generation current. Charges generated in the neutral region move only by diffusion due to the absence of any significant electric field. If this charge is generated near the transition to the depletion region, it can reach the influence of the electric field and will be swept to the opposite electrode, giving rise to the diffusion current.

The total reverse current of a *pn* junction operated under reverse bias can be expressed by [17]

$$J_R = q \cdot n_i^2 \cdot \left(\frac{1}{N_a} \cdot \sqrt{\frac{D_n}{\tau_n}} + \frac{1}{N_d} \cdot \sqrt{\frac{D_p}{\tau_p}} \right) + q \cdot \left(\frac{n_i \cdot W}{2 \cdot \tau_g} \right) \quad (2.12)$$

where D_n and D_p are the electrons and the holes diffusion constants, τ_n and τ_p are the electrons and the holes lifetimes and τ_g is the generation lifetime. Silicon detectors operated under reverse bias are usually fully depleted and the generation current dominates. The total leakage current can be reduced by decreasing the temperature of operation of the detector by a cooling system.

The electric field in the depletion region increases with the reverse bias voltage. There is a critical value of the electric field where the charge carriers are accelerated high enough to ionize atoms of the crystal lattice. The new electron-hole pairs created also gain kinetic energy and participate in the release of more carriers. This avalanche can result in the flow of very large currents which can lead to the breakdown of the diode if the maximum junction temperature is exceeded.

2.3. Principles of operation of silicon detectors

Silicon detectors are based on asymmetric *pn* structures. Usually, the junction is formed by diffusing or ion-implanting a highly doped p^+ thin layer into a lightly doped *n*-type bulk (p^+-*n* detector), although other possibilities exist (like n^+-*p* and n^+-*n* detectors, with another p^+ thin layer under the *p*-type or *n*-type bulk) as

exposed in section 2.5. The depletion region then extends predominantly in the lightly doped bulk, which is used as sensitive detection volume. A highly doped layer of the same type as the bulk is used as back contact to form an ohmic contact. Aluminium metallized layers provide electrical contacts to the highly doped implants that form the electrodes.

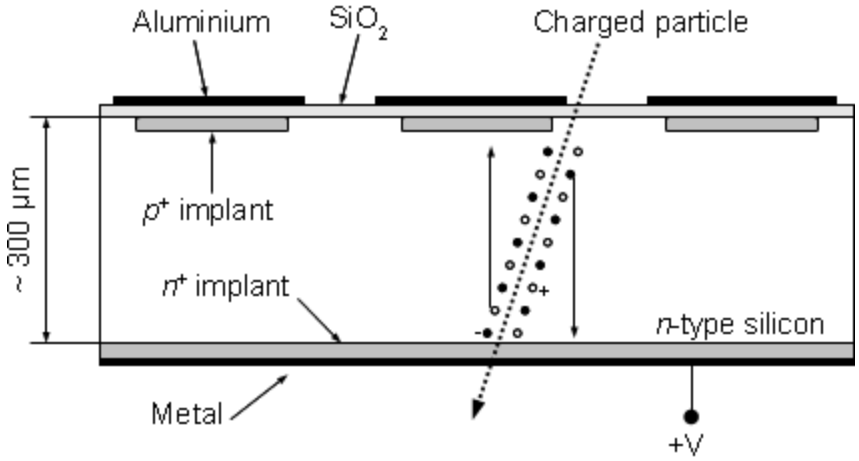

Figure 2.4. *Schematic cross section of an ac coupled p^+-n silicon microstrip detector. Electrons (black circles) drift towards the n^+ back-plane, while holes (empty circles) towards the p^+ implants. An insulator (SiO_2) is used to protect the silicon of the wafer. The strips are connected to the readout electronics through an aluminum layer.*

The principle of operation of a silicon sensor can be summarized as follows:

- an external reverse bias voltage is applied to the sensor junction. For a n^+-p detector, the reverse bias is applied by means of a negative voltage at the lower electrode (p^+ contact). For a p^+-n detector the reverse bias is applied by a positive voltage at the lower electrode (n^+ contact).

- This reverse bias voltage produces a depleted zone (detection volume) free of mobile charges with an electric field.

- When a charged particle traverses the detection volume, it loses energy in ionizing processes resulting in the generation of charge carriers.

- The generated charge carriers drift towards the electrodes due to the influence of the electric field.

- Therefore, a detectable electrical current pulse proportional to the

deposited charge is produced in the electrodes.

The main application of silicon detectors in high energy physics is tracking of charged particles. The electrodes of the sensor can be segmented to form strips or pixels. For accurate position measurements a high segmentation is required. This is achieved by dividing the upper implant (p+ or n+ depending on the detector type) in small regions. In the case of silicon microstrip detectors the upper implant is segmented into shallow electrodes known as strips, as is shown in figure 2.4. However, in the case of pixel detectors the upper implant is segmented into rectangular electrodes of variable size.

2.3.1. Energy loss of charged particles

The mean rate of energy loss of a particle passing through matter is given by the Bethe-Bloch equation [18]. At low particle energies the energy loss is proportional to $1/\beta^2$, at high energies it rises with $ln(\beta^2)$, where β is the velocity of the traversing particle in units of speed of light (v/c). This mean rate for a given material depends only on the charge of the particle, mass and velocity. Above a given velocity, the energy loss reaches a minimum independently of the medium. However the value of this minimum depends on the medium and its density. A particle depositing a minimum of energy per path length is called a minimum ionising particle (*mip*).

This energy loss is of stochastic nature due to variations on both the number of electronic collisions and the amount of energy transferred at each collision. The Landau probability distribution function describes the energy loss in an absorber of a specific thickness [19]. For minimum ionizing particles traversing a 100 μm thick silicon wafer, the predicted most probable energy loss is 26 keV. Since the ionization potential for silicon is 3.62 eV at room temperature (300 K), the number of electron-hole pairs created is 72 per micron of traversed material. Therefore, the most probable charge deposited by a *mip* traversing a typical 300 μm thick is 22500 electron-hole pairs or 3.6 fC.

2.3.2. Signal collection

The electron-hole pairs created in the depletion region after the passage of a charged particle drift in opposite directions to the electrodes of the device under the influence of the electric field. In case of a p^+-n detector (figure 2.4), holes will go to the junction and electrons to the back-plane whereas in a n^+-p detector, electrons will go to the junction and holes to the back-plane.

For electric fields low enough that the velocity change due to acceleration is small compared to the thermal velocity and the mean collision time is independent of the field, the drift velocity of a charge carrier is only a function of electric field and the mobility,

$$\vec{v} = \mu \cdot \vec{E} . \quad (2.13)$$

The mobility depends on the electric field and on the temperature. The mobility of the electrons is higher than the mobility of holes. In silicon at 300 K, the mobility of electrons is 1500 $cm^2/V{\cdot}s$ and the mobility of holes is 450 $cm^2/V{\cdot}s$ [14].

The detector current signal is formed when the carriers start their motion. Once a charge pair is created, the moving carriers, both electrons and holes, couple to the electrodes and induce mirror charges of equal magnitude. The induced charges on the electrodes can be calculated by means of the Ramo's theorem [20]. The instantaneous current induced on a given electrode by a moving charge q can be expressed in terms of the velocity of the charge and a weighting field

$$i = -q \cdot \vec{v} \cdot \vec{E}_w . \quad (2.14)$$

The weighting field is a gradient of a weighting potential, which can be calculated by solving the Laplace equation for the weighting potential considering unity potential on the measurement electrode and zero on all others. The electric field and the weighting field are different for any configuration with more than two electrodes. The electric field determines the charge velocity and trajectory. The weighting field determines how charge motion couples to a specific electrode and depends only on the geometry of the detector. Thus, the weighting field depends strongly on the distance to the strip, only moving charge close to the strip contribute significantly to the signal.

The shape of the current pulse at a given electrode depends on the contribution of both electrons and holes. Electrons and holes move in opposite directions but they have opposite charge, so they induce current of the same sign at a given electrode. The duration of the current pulse is determined by the collection time. The time required for a charge carrier to reach an electrode is called the collection time

$$t_c = \frac{d^2}{2 \cdot \mu \cdot V_d} \cdot \ln\left[\frac{V_{bias} + V_d}{V_{bias} - V_d + 2 \cdot V_d \cdot (1 - x/d)}\right] \quad (2.15)$$

being V_d the depletion voltage, V_{bias} the bias voltage and x the distance where the carrier was created with respect to the readout electrode. For a given bias voltage, the minimum collection time is reached when the carriers are created close to the readout electrode, *i.e.* $x = 0$, and the maximum when the carrier has to traverse the full detector thickness, *i.e.* $x = d$. If the bias voltage is not high enough to fully deplete the detector the collection time tends to infinite and the induced currents will have long tails.

The total induced charge Q at a given electrode will be the sum of the induced currents by the moving electrons and holes

$$Q = \int_0^{t_{ce}} i_e(t) \cdot dt + \int_0^{t_{ch}} i_h(t) \cdot dt \quad (2.16)$$

where t_{ce} and t_{ch} are the collection times for electrons and holes respectively, and $i_e(t)$ and $i_h(t)$ are the induced currents at a given electrode by electrons and holes. The collection of electrons is much faster than the collection of holes (approximately a factor of three) because of their different mobility values. The charge collection time can be decreased by operating the detector at bias voltage exceeding the depletion voltage, also known as *over-depletion* voltage. Nevertheless, care must be taken in order to avoid the breakdown of the junction caused by the avalanching process.

A fully depleted detector volume is very desirable due to various reasons. First, the *Charge Collection Efficiency* (CCE), *i.e.* the signal, is proportional to the depletion depth (W). Second, the noise is proportional to the detector capacitance C_d, which is proportional to $1/d$ (equation 2.12). Third, the resolution is decreased by charge spread in a partially depleted detector volume.

2.3.3. Position resolution

The charge carriers follow the electric field lines in their average drift movement. However, they are also subject to thermal diffusion which originates the spread of the charge cloud transversely to the drift path. Longitudinal diffusion, due to carriers having drift velocities greater or less than the average, gives a spread in drift times. The charge distribution is described by a Gaussian law and the standard deviation σ of the distribution is given by

$$\sigma = \sqrt{2 \cdot D \cdot t_c} \quad (2.17)$$

where D is the diffusion coefficient and t_c is the charge collection time. The

diffusion coefficient is given by the Einstein relation

$$D = \frac{k \cdot T \cdot \mu}{q}. \quad (2.18)$$

The transverse diffusion is the same for holes and electrons, since the the collection time is inversely proportional to the carrier mobility.

The position resolution of a silicon strip sensor depends mainly on the geometry, as the width of the electrodes and the pitch (distance between electrodes or strips), the diffusion width of the charge cloud and the method of readout. Using binary readout (it is only distinguished if a strip is hit or not), the position resolution, Δx (root mean square deviation from the true coordinate) is given for a strip pitch p by

$$\Delta x = \frac{p}{\sqrt{12}}. \quad (2.19)$$

In practice, the readout electronics sets the constrains on the width and pitch of the detector strips. Typical strip pitches are 20−200 μm, which results in resolutions of approximately 6−60 μm.

The transverse diffusion can improve the intrinsic resolution of the detector, since the charge is spread to adjacent strips. Therefore, the collected charge is shared between more readout electrodes. If the strip pitch is chosen small enough so that the signal charge is collected on more than one strip, the measurement precision can be substantially improved with an analogue readout. The coordinate is found by interpolation, *i.e.* by the centre of gravity of the input charge cloud. The position resolution, Δx, is then limited by the *signal-to-noise (S/N)* performance of the electronics and the readout strip pitch [16]

$$\Delta x \approx \frac{p}{(S/N)}. \quad (2.20)$$

The range of charge interpolation can be extended over distances larger than the diffusion width by capacitive charge division [16, 17].

2.4. Radiation damage

The detailed explanation of the radiation damage and effects on silicon sensors is beyond the scope of this thesis but some effects and mechanisms can be

summarized. Basically, three main macroscopic radiation effects can be mentioned [21, 22]:

- defects in the silicon bulk induced by radiation produce severe changes in the doping characteristics of the bulk material, resulting in donor removal and the creation of acceptor like defect centres. Over an irradiation fluence value, there will be a type inversion of the bulk material, changing from *n*-type to *p*-type.

- An increase of the leakage current due to defect centres in the silicon bulk induced by radiation. These defect centres introduce new energy levels in the forbidden silicon energy gap.

- A decrease on the charge collection efficiency due to charge trapping induced by charge trapping centres (*i.e.* shallow levels in the silicon energy gap band induced by radiation).

Two basic radiation damage mechanisms affect silicon detectors [21, 22]. The first one is the bulk damage. High energetic hadrons can suffer scattering collisions with the silicon atoms of the lattice. In this case, the *non-ionizing energy loss* (*NIEL*) is at the origin of the damage bulk silicon sensors. Displacement damage occurs whether the energy transferred to the silicon atom is sufficient to remove it from the crystal lattice. The atom is then called primary *knock-on atom* (*PKA*). The recoiled silicon *PKA* loses energy by means of two competing processes, ionization and further collisions, producing more lattice displacements (vacancy-interstitial or *Frenkel* pairs). The second mechanism is the surface damage. The damage by ionizing radiation is mainly induced by the accumulation of charges in the interface of the silicon dioxide (SiO_2) and the silicon bulk, where there is a lattice mismatch. The charge carriers created in the depletion region can be trapped in the $SiO_2 - Si$ interface. The probability of trapping is higher for holes than for electrons, due to the difference in their mobility values. Hence the charge accumulated at the oxide-silicon interface is mainly positive signed.

The displacement damages in the material lead to defects states in the silicon bulk which create thermal excitations and enable a possible charge carrier transfer to the conduction band. Hence, the leakage current of the detector increases. The correlation between the leakage current and the radiation damage is based on the assumption that the measured current is dominated by the bulk generation current and then it is proportional to the density of radiation induced defects and to the depleted volume.

Additionally, the defect complexes also act as recombination centres which result in a decrease of the output signal of a silicon sensor. A possible change in the doping concentration will also affect the depth of the depletion area or increases the full depletion voltage. The increase of the leakage current of the sensor results in an rise of the total power consumption, a growth of the shot noise and/or an upsurge of the *dc* signal current that bypass the *ac* coupling.

2.5. Silicon detector structures

In radiation detection applications, the usage of silicon detectors has some advantages over other types of radiation detectors. First, the high density of the silicon compared to any gas allows having good detection efficiency in thin detectors. Second, the energy necessary to generate an electron-hole pair in silicon (3.6 eV) is a factor of ten lower than in gas detectors and hundred times lower than in a scintillator, leading to a better energy resolution. Third, the time resolution is higher due to a brief charge collection time. Finally, silicon detectors can be designed with a high segmentation, having a very good spatial resolution.

On the other hand, there are some disadvantages when using silicon detectors as radiation detectors. Firstly, silicon detectors have high conductivity compared to gas detectors. Therefore, there will be a background noise in silicon detectors that will tend to affect the detection of very low energy ionizing particles. Secondly, the defects in the silicon crystal lattice will produce recombination of charge carriers decreasing the collection charge efficiency. There are three main types of silicon sensors according to the silicon detector structure:

- ***n*$^+$-*p*:** this type of detector is formed by a high resistivity *p*-type substrate and n^+ implants over the substrate. It has a p^+ implant at the bottom more doped than the substrate in order to act as a contact with the lower metallic layer. The main advantage of this type of silicon detector is that the collection time is low because electrons are collected which have higher mobility than holes. However, p^+ implants between the n^+ implants are needed to isolate them. The manufacturing process of this type of detectors is complex. This type of detector can operate partially depleted, since the depletion region grows from the n^+ implants to the rest of the detector.

- ***p*$^+$-*n*:** the detector is composed of a high resistivity *n*-type substrate and p^+ implants over the substrate. It has a n^+ implant at the bottom more doped than the substrate in order to act as a contact with the lower metallic layer. The charge collection time in this case is lower because holes are collected, as well as the radiation hardness. However, the manufacturing process is

simpler. This type of detector can also work partially depleted as the n^+-p detectors, although the depletion volume grows from the n^+ back-plane.

- ***n*⁺-*n*:** the detector has a *n*-type substrate and n^+ implants over the substrate. The bottom implant is a p^+ material. In this case electrons are collected, so the charge collection time is low. Furthermore, the radiation hardness is the highest. The technology of this type of detectors is much more complex. These detectors can operate after type inversion, *i.e.* after being highly irradiated, partially depleted. Before type inversion, as the depletion junction grows from the p^+ side, the detector must operate fully depleted. After type inversion, however, the depletion junction will shift to the n^+ side. In that case the detector can be read out even if partially depleted, being able to operate at a reasonably low bias voltage even in the case of a large depletion voltage.

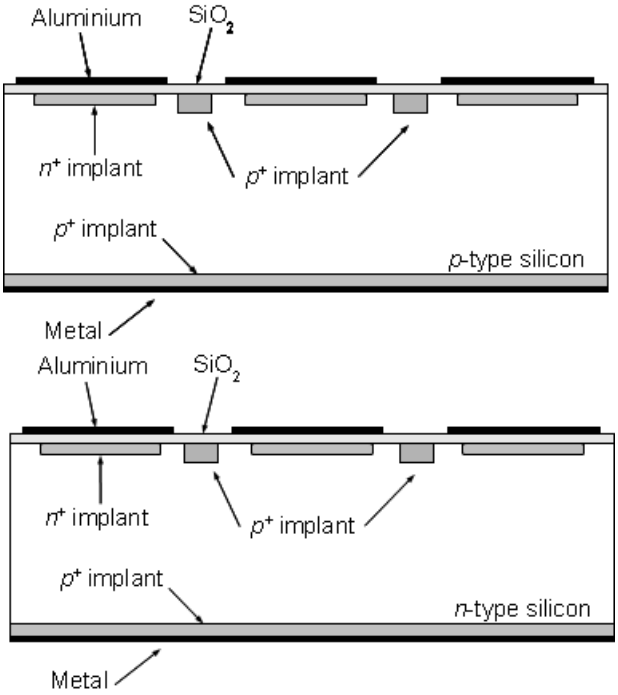

Figure 2.5. *Structure of n^+-p and n^+-n silicon detectors respectively (from top to bottom).*

The structure of a p^+-n detector is shown in figure 2.4 while the structure of a n^+-p and a n^+-n detector is shown in figure 2.5. For instance, ATLAS microstrip detectors used in the Semiconductor Tracker (SCT) are p^+-n detectors whereas the ATLAS silicon pixel sensors used in the Pixel Detector are based on n^+-n

technology.

2.6. Signal acquisition

The current pulse detector signal readout is carried out, firstly, by means of an amplifier because this signal is generally of small amplitude. A typical output pulse for a *mip* traversing a 320 μm p^+-*n* sensor fully depleted is shown in figure 2.6. The signal duration is about 0.1-30 ns for a 10-300 μm thick detector determined by the collection time. The signal has also a low rise time due to diffusion [24, 25]. The total charge contained in the current pulse is proportional to the energy deposited in the detector by a charged particle traversing it (3.6 fC for a *mip* traversing a 300 μm thick detector), as it has been explained in section 2.3.1.

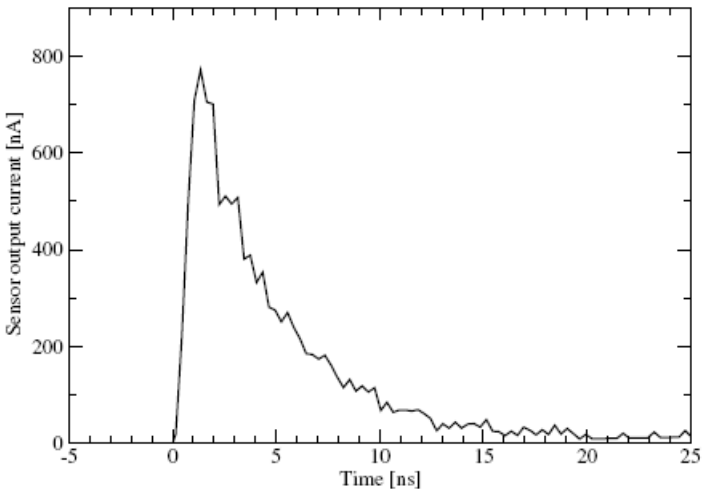

Figure 2.6. *Real primary current pulse of a silicon sensor. The pulse corresponds to a mip traversing a p^+-n sensor with a width of 320 μm. The p^+ implant width is 60 μm, the electrode metal width is 68 μm and the strip pitch is 198 μm. The strips have a length of 108 mm. The sensor is biased with 80 V and has a full depletion voltage of 70 V. Figure taken from [23].*

2.6.1. Amplification

The equivalent circuit of a microstrip silicon detector can be represented as a current source in parallel with a capacitor as shown in figure 2.7. The current source supplies the current pulse when generated charge is moving in the sensitive detector volume. The capacitor represents the total capacitance of the detector. In a

detector diode is the junction capacitance given by equation 2.9. In a microstrip silicon sensor the total strip capacitance should be considered, which is the combination of the inter-strip capacitances to the neighbouring strips and the capacitance to the back-plane. The inter-strip capacitance is usually about 1-2 pF/cm for strip pitches of 25-100 μm. The back-plane capacitance is typically 20 % of the inter-strip capacitance [17].

A decoupling capacitor must be used in order to minimize the leakage current in the detector signal readout. This capacitor can be integrated in the detector structure, capacitive coupling, or placed externally, direct coupling. In case of capacitive coupling the capacitor is created between the implant and the electrode, using SiO_2 as insulator. When there is a direct coupling, an external capacitor is placed between the electrode and the input of the amplifier. The coupling capacitance is typically much higher than the inter-strip capacitance.

Three basic amplifier configurations are used for reading the output signal of microstrip silicon detectors [17, 26, 27]:

- the **charge sensitive amplifier (CSA)**, which actively integrates the detector current on the amplifier feedback capacitor. This is the best configuration concerning noise. In addition the CSA features a purely resistive input impedance and avoids any dependence on the change of the detector capacitance, C_d, with temperature.

- The **voltage amplifier**, where the signal current is integrated on the detector capacitance itself. The voltage across the capacitor is amplified. Since the amplifier output voltage depends on the detector capacitance C_d, this configuration is only of interest if C_d is constant, *i.e.*, the depletion depth is constant and the strip lengths are the same for all sensor strips. Temperature fluctuations and radiation damage can lead to a variation of C_d. Beside this, the voltage amplifier is more noisy than the charge sensitive loop.

- The **current amplifier**, the signal current is directly amplified and converted into a voltage. The main disadvantage of this loop configuration is the inductive input impedance of the configuration for frequencies higher than the cutoff frequency of the amplifier, which limits the use with large capacitive loads. The current amplifier is also noisier than the CSA.

Since the CSA amplifier is the configuration integrated in the readout ASIC used in this system, its principle of operation will be explained. A schematic view

of such type of amplifier is shown in figure 2.7. There is a voltage inverter amplifier with a high gain -*A*. Its output is fed back to the input across capacitance *Cf* in parallel with a resistor *Rf* of high value (in the order of MΩ). The resistor provides a *dc* return for the bias currents of the input device. If *Cf(1+A)* is chosen high enough with respect to the detector capacitance, high efficiency charge acquisitions are achieved. The feedback resistor *Rf* discharges the feedback capacitor *Cf* after each integration. Considering a very fast rise time for the amplifier, the response is given by

$$V_{out} = \frac{-Q_s}{C_f} \cdot e^{-t/\tau_f} \quad (2.21)$$

where Qs is the input charge. The charge gain is determined by, the feedback capacitor, a well controlled component. The resulting pulse has a long exponential tail determined by the time constant $\tau_f = R_f C_f$ and ranges from a few μs to 100 μs. The large time constant of the tail causes the problem of pile-up: if further input signals arrive within τ_f, their output signal will add to the tail of the predecessor pulse, which produces distortion.

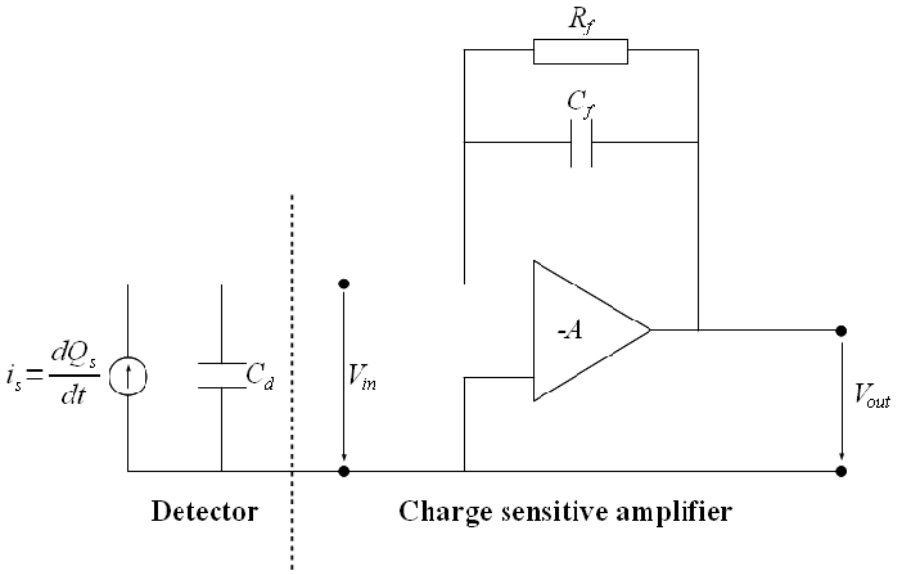

Figure 2.7. *Schematic of a charge sensitive amplifier (amplifier with gain -A) with a feedback capacitor Cf and a feedback resistor Rf, connected to a silicon detector with a capacitance Cd and providing a current pulse is.*

A very important characteristic of the CSA is that its input impedance can be

made resistive for the frequencies of interest for silicon detectors. For frequencies higher than the cutoff frequency of the amplifier the input impedance of the CSA becomes resistive up to the bandwidth of the amplifier, $\omega 0$, and is given by

$$Z_i = R_i \approx \frac{1}{\omega_0 \cdot C_f}. \quad (2.22)$$

This resistive term can be less noisy than a physical resistor of the same value. The sensor capacitance is discharged by the resistive input impedance of the CSA with a time constant

$$\tau_i = R_i \cdot C_d = \frac{C_d}{\omega_0 \cdot C_f}. \quad (2.23)$$

This time constant determines the rise time of the CSA, which increases with the detector capacitance (or other parasitic capacitances at the input of the amplifier).

2.6.2. Pulse shaping

Shaping the amplifier output has two main reasons: pile up prevention and signal-to-noise optimization. The pile-up effect can be avoided by shortening the pulse tail by shaping. A restriction of the signal bandwidth, also by shaping, broadens the pulse and results in an improved signal-to-noise ratio. These two objectives are in conflict, the first one aims at decreasing the pulse width, the second one at increasing it. For a certain application also the dynamic range and power consumption have to be considered for the shaper design.

There are two types of shapers [17, 24]: time-variant shapers and time-invariant shapers. The former ones can vary their characteristic response with time according to, for instance, the output pulse height or with an external gate signal. The latter ones have a fixed characteristic response with time. Since a CR-RC time-invariant shaper is implemented in the readout ASIC used in this system, this type of shapers will be explained with more detail.

The *CR-RC* pulse shaping is a widely used technique. The frequency spectrum is restricted by low-pass filtering with a differentiation stage (*CR*) and by high-pass filtering with a integration stage (*RC*). The form of the output pulse is called semi-Gaussian. The high-pass filter sets the duration of the pulse by introducing a decay constant, τ_d, and the low-pass filter increases the rise time by means of the integration constant, τ_i. For a given differentiation time, the CR-RC shaper have the

optimum signal-to-noise ratio when $\tau_d = \tau_i = \tau$. In this case, the peaking time, when the output pulse reaches its maximum value, will be τ.

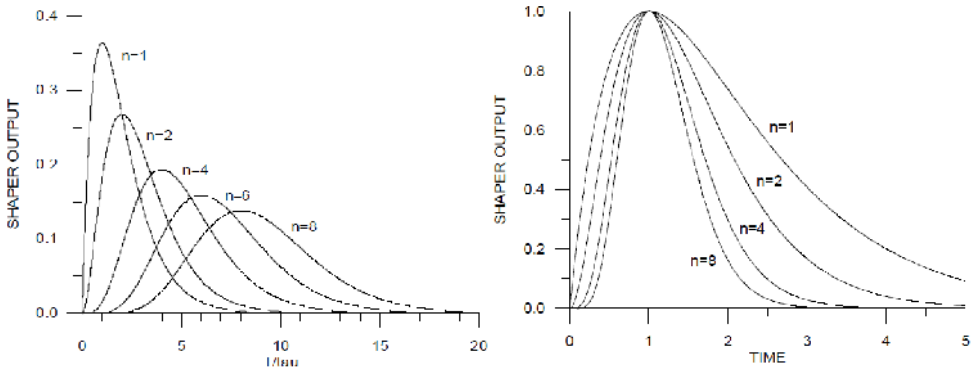

Figure 2.8. *Output response of shapers with multiple integration stages. In the left figure the integration constant is the same for all integration stages. In the right figure the integration constant of subsequent integration stages is varied according to equation 2.25. Figure from [28].*

A theoretical 18% improvement of the signal-to-noise ratio with simple *CR-RC* shaping can be achieved if the pulse has ideal Gaussian form. Since a Gaussian shaper is not realizable electronically [18], with a network of one *CR* differentiation stage and a series of many *RC* integrating stages (*CR-RC*^{n}) a pulse can be produced which is close to a Gaussian shape. The order of the shaper is denoted by *n*. In figure 2.8 it is shown the output response of shapers with multiple integration stages. In the left side figure, all integration stages have the same time constant $\tau_i = (RC)_i$, whereas in the right side figure the peaking time is constant. This is achieved by varying the integration time constant of subsequent stages according to

$$\tau_{n+1} = \frac{1}{n+1} \cdot \tau_n . \quad (2.24)$$

The pulses are more symmetrical improving the rate capability at the same peaking time. The disadvantage of Gaussian pulses is their larger width compared to *RC* shaped pulses, which can cause overlap problems at high counting rates. The peaking time of the shaper has to match the primary signal width. Otherwise a loss in signal amplitude occurs, which is called ballistic deficit. With $n = 1$ the simplest *CR-RC* shaper configuration is obtained. Figure 2.9 and figure 2.10 show the passive and active implementations of a *CR-RC* shaper, respectively.

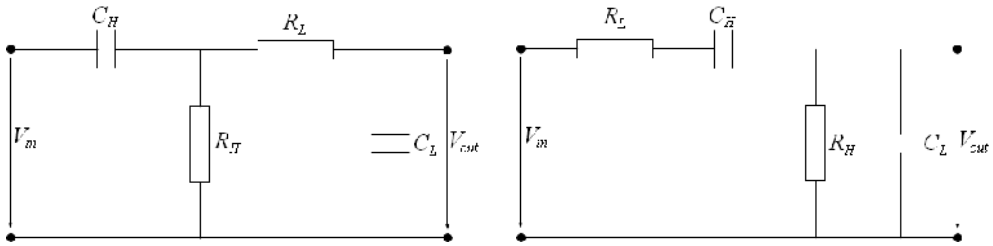

Figure 2.9. *Passive implementations of a CR-RC shaper.*

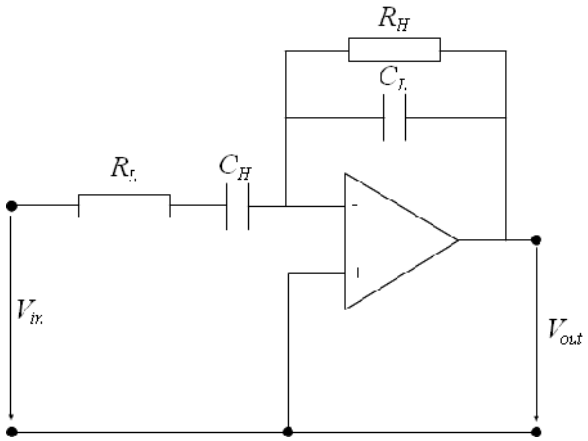

Figure 2.10. *Active implementation of a CR-RC shaper.*

2.7. Noise

The signal of a silicon detector is disturbed in both amplitude and time by superimposed noise. The fluctuations in the energy loss produce a statistical variation in the charge collected. Noise is also introduced by the readout electronics affecting the charge measurement or the biasing source. Therefore, the *signal-to-noise ratio* gets degraded.

The main sources of noise arise from velocity and number fluctuations of the charge carriers contributing to the signal current of the detector. Random velocity fluctuations due to thermal excitation are superimposed to the average drift velocity. This *thermal noise* (also called *Johnson noise*) has a white spectral density, i.e. the noise power per bandwidth unit is constant. Fluctuations in the number of charge carriers occur in the current flow. There is a non-constant current due to these fluctuations as well as a constant detector leakage current corresponding to the mean flux of charge carriers. This noise is known as *shot noise*, which also has a white spectrum.

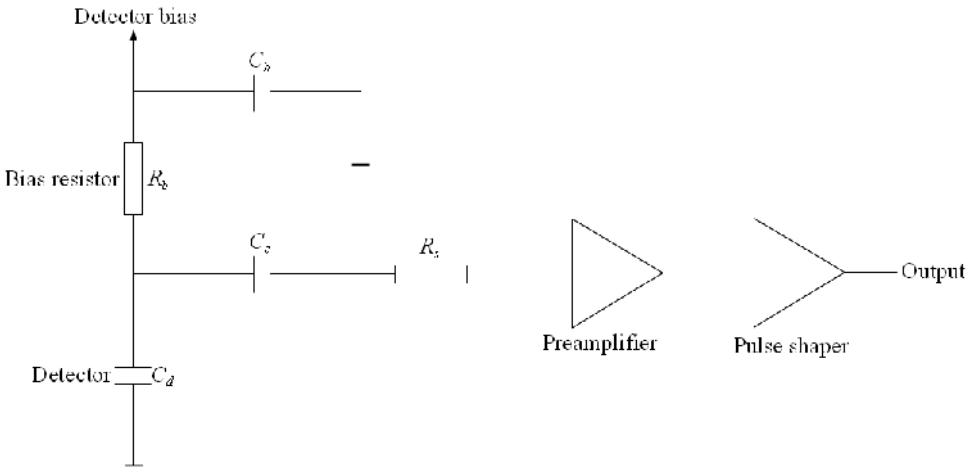

Figure 2.11. *Schematic of a typical front-end circuit for a silicon detector.*

The noise contributions in a silicon detector can be determined from the analysis of the front-end circuit (figure 2.11). The sensor is represented by a capacitance C_d, the bias voltage is supplied through a resistor R_b and the detector is coupled to the preamplifier through a *dc* blocking capacitor C_c. The series resistance R_S is the sum of all resistances present in the signal path, the electrode resistance, any protection networks and parasitic resistances in the preamplifier input transistor.

The noise can be described in terms of either voltage or current sources. The shot noise i_{nd} of the detector leakage current and the thermal noise i_{nb} of the bias resistor are represented by current noise generators. The thermal noise due to the series resistor R_S acts as a voltage generator (e_{ns}). The amplifier white noise is described by a combination of voltage (e_{na}) and current noise sources (i_{na}) at its input. The amplifier has also a excess noise represented by a voltage source $e_{nf} = \sqrt{A_f/f}$. This noise has dependency with the frequency of the type *1/f* and it is called *Flicker noise*.

The noise is expressed in terms of the equivalent noise charge (*ENC*), corresponding to a signal that would generate an output voltage of the same value as the rms (*root mean square*) of the noise distribution in the system, *i.e.* signal-to-noise ratio equal to one. The *ENC* is expressed in Coulombs or the corresponding number of electrons. The total noise is given by the individual noise contributions added in quadrature, which can be expressed as [17]:

$$(ENC)^2 = F_i \cdot T_s \cdot \left(2 \cdot q \cdot I_d + \frac{4 \cdot k \cdot T}{R_b} + i_{na}^2\right) + \frac{F_v \cdot C_d^2}{T_s} \cdot \left(4 \cdot k \cdot T \cdot R_s + e_{na}^2\right) + F_{vf} \cdot A_f \cdot C^2 \qquad (2.25)$$

where T_s is the shaping time, F_i, F_v, and F_{vf} depend on the shape of the pulse determined by the shaper and I_d is the leakage current of the detector. The voltage noise contributions increase with the capacitance. The voltage noise dominates with short shaping times whereas the contribution of the current noise increases with the shaping time. Noise with *1/f* spectrum depends only on the ratio of the upper and lower cutoff frequencies of the shaper. The *ENC* can be minimized by choosing an appropriate value for the shaping time, optimizing the signal-to-noise ratio.

Chapter 3

Specifications and architecture of the system

In this chapter the specifications and the architecture of the readout system for microstrip silicon sensors are detailed. In section 3.1, there is a brief description of the experimental methods for determining the main parameters of the silicon sensors. First, the experimental methods which do not need signal generation are described, i.e. C-V and I-V measurements. Then, the setups used for characterizing the silicon sensors in which signal generation, i.e. electron-hole pairs, is necessary are explained. Among these experimental methods, the radioactive source setup and the laser setup are key issues for designing the system. Therefore, a discussion about the advantages and the disadvantages of the current acquisition systems used with these setups is done. The motivations for designing the readout system are derived from this discussion. Once the motivations are stated, the specifications of the system are listed in section 3.2. Finally, the system architecture designed to fulfill these specifications is described in section 3.3.

3.1. Experimental methods

As explained in the first chapter, there is a need of studying the main properties of highly irradiated microstrip silicon sensors. This type of detectors are currently being used at the LHC experiments. Since a higher luminosity is intended to be achieved at HL-LHC experiments, it is required to predict the behaviour of this

kind of sensors after a high irradiation dose. There are basic detector parameters which need to be determined like the leakage current, the full-depletion voltage, the breakdown voltage, the effective charge concentration, the charge collection efficiency, the timing resolution or the position resolution. These parameters must be assessed by means of experimental methods before and after irradiation of the detector. A number of these methods are detailed in [21, 29].

3.1.1. C-V and I-V measurements for silicon sensors

Different electrical detector parameters can be determined without the need pf generating signal in the detector. For instance, the full-depletion voltage, V_{FD}, and the effective charge concentration N_{eff}, can be determined by means of capacitance-voltage measurement (figure 3.1). For this kind of measurement either a capacimeter, and an *AC* voltage source, or a LCR meter are required.

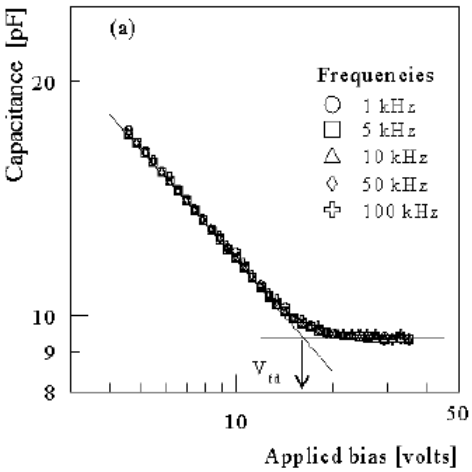

Figure 3.1. *Typical C-V characteristic with the voltage frequency as a parameter for a non-irradiated silicon sensors. The intersection of the interpolated slopes gives the full-depletion voltage. Figure taken from [30].*

The capacitance is calculated from the measured amplitude and phase shift of the current signal in response to an *AC* measuring voltage with user selected frequency. The frequency range used for measuring is important because the capacitance value obtained can vary from the true value for frequencies over 1 MHz or below 1 kHz. The *AC* voltage amplitude should be high enough to have a good signal to noise ratio but low enough not to disturb the measurement at low bias voltages. The data can be represented as a *C-V* curve, in log-log scale, or a $1/C^2$-*V* curve. V_{FD} can be estimated from these curves as the intersection of two lines, one obtained by

interpolating the slope at large depletion depths and the second fitted to the plateau at full depletion. N_{eff} can be determined from V_{FD} according to equation 2.10, if a homogeneous distribution of N_{eff} is assumed.

The leakage current behaviour and the breakdown voltage can be determined by means of a leakage current-reverse bias voltage measurement (figure 3.2). With this type of curves interstrip resistivity measurements on dedicated test structures can be carried out as well for evaluating different interstrip isolation methods [31]. A voltage source with current measurement capability is required for obtaining this kind of measurement. In addition, specialized probe equipment for semiconductors characterization is used for these measurements, in order to connect the voltage sources and the capacimeter (or LCR meter) to the detector.

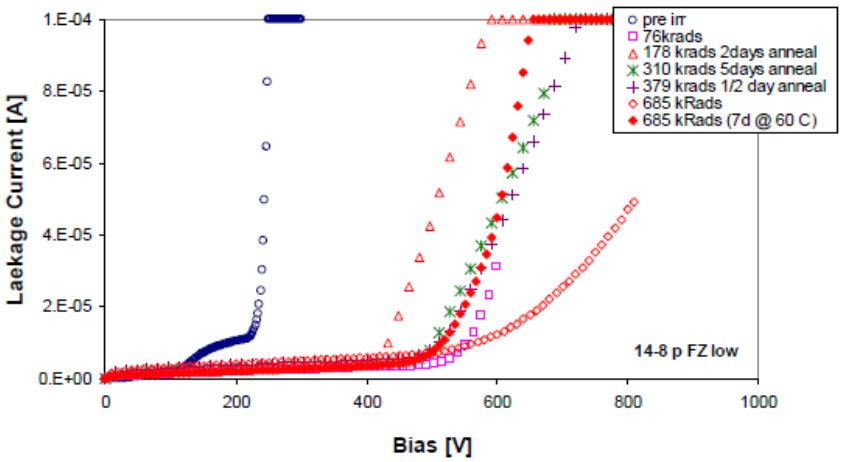

Figure 3.2. *Typical I-V characteristic of a silicon sensor for several irradiation stepsand annealing times. Figure from [31].*

3.1.2. Test beam setup

Other important silicon detector parameters, like the charge collection efficiency, the position resolution or the timing resolution, are related to the signal generation and collection in the detector volume when crossed by charged particles. Hence, an external source for the charged particle generation is required in order to characterize these parameters. Moreover, an electronic system for the signal acquisition and processing as well as the triggering will be required.

Currently, there are several ways of characterizing these parameters. A high-energy particle beam could be used in a so-called test beam. Here, the detectors

operate in very similar conditions to the future use of the sensors (for instance, in a LHC experiment). In this case the particle energy is homogeneous and the tracks of particles can be precisely determined by telescopes without compromising the beam quality. A telescope is a tracking system used to measure the spatial resolution performance when developing new detectors. It usually has several parallel planes with microstrip silicon detectors placed perpendicular to the beam for track reconstruction. The detector under test is also placed perpendicular to the beam in a centered position with respect to the tracking planes. Other detectors can be included in the tracking planes for obtaining a common trigger signal for the acquisition system. The acquisition system must read the data from the detector under test, the track detectors (x and y coordinates) and the trigger detectors.

Therefore, both spatial and timing resolution can be evaluated as well as the charge collection efficiency. However, beams of relativistic particles are only available at large accelerators. and usually only at certain periods, so they are very expensive. It is then highly desirable to find cheaper and more readily available alternatives.

3.1.3. Radioactive source setup

One of these alternatives is a radioactive source, which is used in many laboratories. The main problem of this type of source is that charged particles are generated randomly. For instance, a $^{90}Sr/^{90}Y$ source emits β particles (electrons) with energies up to 2.27 MeV [32], which penetrate deeply into the detector or even pass through it. Therefore, they interact similarly to minimum ionizing particles (*mips*, see section 2.3.1) although their energy spectrum is broad and it also produces a large spread of energy deposited in the sensor. Their interaction point cannot be determined by telescopes, which exclude this alternative for position-sensitive measurements. Furthermore, a trigger detection system is necessary in order to provide timing information abut the passage of a particle through the detector.

A typical β source setup diagram is shown in the figure 3.3. The microstrip silicon detector is placed under the β source in order to receive the electrons generated by the source. The reverse bias voltage applied to the detector is supplied by an external voltage source. Two scintillators are situated under the detector as well. In a scintillator, the incident ionizing radiation interacts with the scintillator material and produces secondary electrons (δ-*electrons*). These electrons excite the scintillator atoms and flashes of light (*UV* or visible) are produced when the atoms de-excite. Each scintillator is connected to a photomultiplier tube (*PMT*) in order to transform the light signal of the scintillator into an electrical negative analogue

pulse.

The analogue output signals of the photomultipliers are discriminated, obtaining digital pulses. These pulses are usually connected to a coincidence stage for generating a trigger signal. Another alternative is to use just an output of one discriminator as trigger signal although more false triggers are obtained in this case depending on the photomultiplier noise. For β particles, using one discriminator is sufficient since this type of particles are stopped in the first scintillator. This trigger signal informs when an electron has passed through the detector and the scintillator(s) so is used to trigger an oscilloscope or a data acquisition system. The detector signal is amplified by a current-to voltage amplification stage and connected to the oscilloscope or the data acquisition system. Using an oscilloscope is a simple way to acquire the output signal. The main disadvantage of this kind of acquisition system arises from the fact that just one channel of the detector can be acquired (usually the sum of various channels is also acquired).

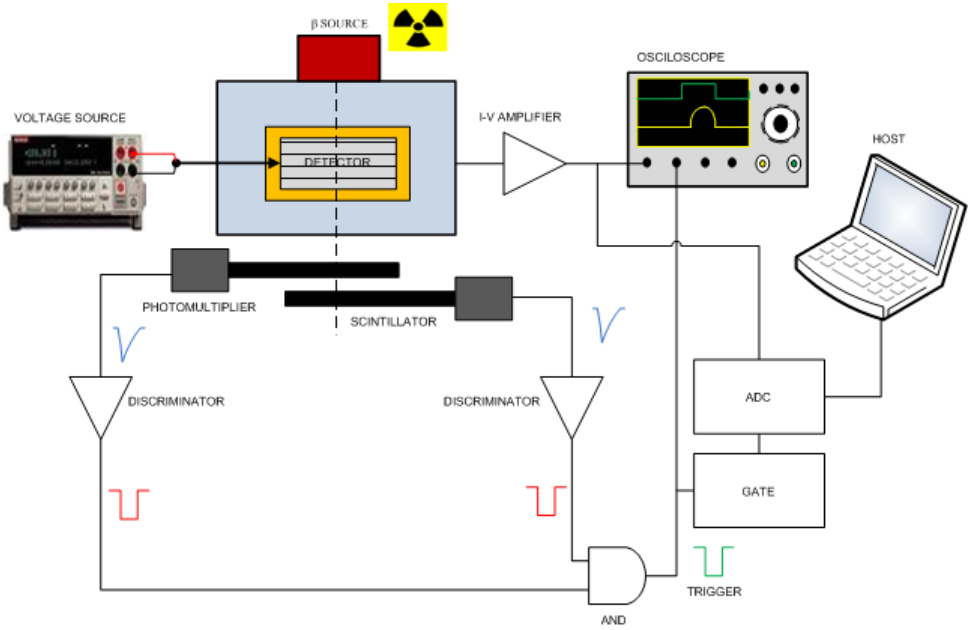

Figure 3.3. *Block diagram of a typical β source acquisition setup.*

3.1.4. Laser setup

Another alternative to induce charge in the detector is a laser light [33, 34]. In this case, a laser pulse from a laser is generated after receiving a pulsed trigger

signal. Red (670 nm of wavelength) and infrared (1060 nm of wavelength) lasers are usually employed. The light is then lead through an optical fibre with a focusing lens at the end and it is projected on to the silicon detector. The detector is placed on a stage moving in three dimensions with micrometer precision. The whole setup is in a light-tight box. This alternative offers a good spatial resolution with a laser beam well focused (small spot diameter of few μm) and good timing resolution (short laser pulse of few ns).

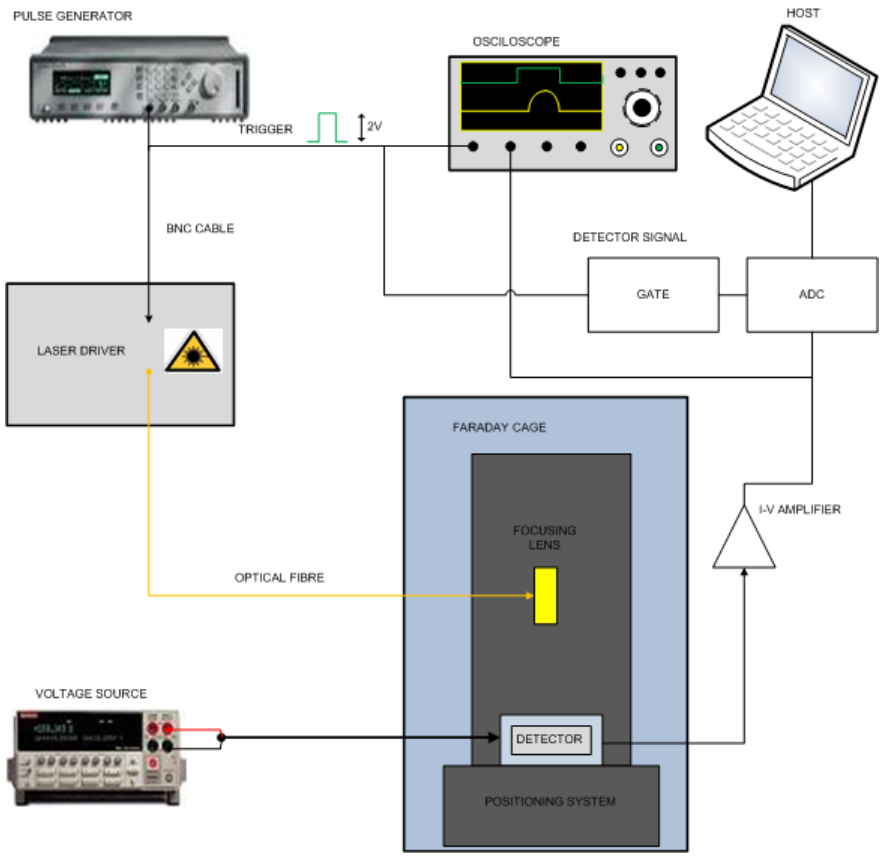

Figure 3.4. *Block diagram of a typical laser acquisition setup.*

The laser beam is a beam of photons instead of charged particles. For red and infrared lasers the penetration and interaction mechanisms of photons differ in several aspects from charged particles. The electron-hole pairs are generated by means of the photoelectric effect. If the photon energy is greater than the band gap energy of silicon, an electron from the valence band will have high probability of being excited and lifted to the conduction band, leaving a hole in the valence band

[14]. The photon energies in red (E_{red}= *1.85 eV*) and infrared ($E_{infrared}$= *1.17 eV*) lasers are higher than the energy gap of silicon (E_g = *1.12 eV*).

With a laser beam of finite width and certain penetration depth, the charge is generated within the full volume traversed by the laser beam. There are no δ-electrons created, which normally increase the cluster size and deteriorate the sensor spatial resolution. On the other hand, the laser light is subject to reflection and refraction on surface and subsequent layers (pads, passivation, protection, etc.), complicating assessment of energy deposited in the sensor. Using an infrared laser of 1060 nm of wavelength is very attractive since the penetration depth of this light is about 1 mm at 300 K in silicon, so the charge induced in a 300 μm thick detector is roughly uniform.

The diagram of a typical laser setup is shown in figure 3.4. In this case, a pulse generator produces a square pulse signal. The generated pulsed signal is used to drive a laser source and also as a trigger signal for the oscilloscope or data acquisition system. This signal usually has a frequency of the order of a few kHz. The laser output of the laser source is an optical fibre which is connected to a focusing lens to project light on the detector. The reverse bias voltage applied to the detector is supplied by an external voltage source, as in the case of the β source setup.

The output signal of the detector is connected to a current-to-voltage amplifier and the output of this stage is connected to an oscilloscope or data acquisition system. The main disadvantage is the same as the β source setup, just one channel of the detector can be acquired (or the sum of various channels). On the other hand, the detector output signal in this setup can be higher than in the β source setup (the laser intensity can be modulated), which means that a lower amplification factor is required at the amplification stage (usually a factor of 100 in this case compared to a factor of 1000 in the case of a radioactive source setup).

3.1.5. Acquisition systems for silicon sensors

The acquisition of the detector signal with the setups explained above is difficult due to different reasons. The main difficulty is reading out all the channels of a microstrip silicon sensor at the same time. In these setups just one detector channel or the sum of various channels (or all) is usually acquired. For instance, in order to read out all the channels of a silicon detector of 128 strips, an amplification stage would be needed for each channel, i.e. 128 amplification stages. Furthermore, the detector output signal, due to its characteristics, requires a fast current-to-voltage amplifier with low noise, high gain factor and high bandwidth. This kind of

amplifier module is expensive since its requirements are demanding.

If a more sophisticated acquisition system is necessary, other type of electronic modules will be required. For instance, an amplifier, an analogue-to-digital-converter (ADC) and a communication controller for sending the data to a host computer will be needed, so that the data acquired from a detector channel can be stored in a computer for further processing. Moreover, additional electronic modules as discriminators, gate generators or time-to-digital-converter (TDC) modules will have to be used for triggering purposes. Then, the cost of the acquisition setup will increase considerably either if commercial electronic modules or custom electronics are used following this philosophy. In addition, there will be a minimum standardization as often the same functions are required (amplification, signal conditioning, ADC, computer communication, temporal logic, discrimination, etc) but there will be implemented with different modules (VME, CAMAC, NIM or custom electronic modules) depending on the laboratory facility. Hence, it will be more difficult to obtain comparable measurements in different laboratories.

On the other hand, it would be interesting to test silicon sensors with an electronic system as similar as possible to those used at real experiment (e.g. LHC experiments). An option would be to use an electronic acquisition system already developed for any LHC experiment. The disadvantage of such an electronic system comes from the fact that it is designed to operate in an environment quite different than these laboratory setups. However, for a test beam it could be used as beam telescope without major problems as reported in [35-38]. Other telescopes based on custom electronics have been developed for test beam purposes [39, 40]. The main disadvantage of the telescope systems is that they are specifically designed for test beam and they are not easily adaptable for the aforementioned laboratory setups. Furthermore, the cost of these systems is very high. Hence, an optimized electronic system for these laboratory setups and easily upgradable for a test beam setup would be an advantage.

Taking this into account, a front-end readout chip used at the LHC experiments could be a good choice in order to be used in a readout system of this type. This kind of readout chip is specially valuable since it will allow to read many detector channels in parallel. An analogue measurement of the pulse shape with this kind of readout chips would be particularly interesting because it would allow to carry out more reliable charge collection research than with a binary measurement. Therefore, a front-end readout chip with analogue output capability should be used. In addition, these front-end chips are usually radiation hard which is an advantage because they can be placed close to the silicon detector.

3.2. System specifications

The main purpose of the ALIBAVA collaboration, as it has been explained in chapter 1, is to design a readout system for microstrip silicon sensors which could be used with different laboratory setups for researching some important parameters of these detectors. This system must be able to measure the collected charge in microstrip silicon sensors by reading out all the channels of the detector at the same time as an analogue measurement. For this, a front-end readout chip used in any of the LHC experiments should be used. Thus, it will be possible to have an acquisition system as similar as possible to those used in the real LHC experiments for reading out silicon sensors. On the other hand, this system will be used with a laser setup and a radioactive source setup, so its design must be optimized for the operation with these kinds of setups.

In order to fulfill these requirements the system should meet the following specifications.

- First, the system must be compact and portable in order to install it easily at any laboratory.

- Second, the system will contain two specific front-end readout chips for reading out microstrip silicon sensors in analogue mode. The front-end chips used will be the Beetle readout chips [41]. These chips are used in the Vertex detector, the Silicon tracker detector and the RICH detector of the LHCb experiment. The Beetle chip has been chosen since it fulfills the requirements about radiation hardness, analogue readout and it is already being used in a LHC experiment. The characteristics of the Beetle readout chip will be explained with more detail in the next chapter.

- Third, the system will be used with two different laboratory setups, the laser setup and the radioactive source setup. The system will manage the trigger signals in both cases. For the radioactive source setup, it will have trigger inputs for two photomultipliers and for an auxiliary signal. In particular, the system will incorporate two inputs for analogue signals coming from photomultipliers. The auxiliary input can be both a positive or negative pulsed signal, either in voltage or current. The usage of these signals will be user selectable. The system will have a discrimination stage and the coincidence required for the analogue signals. For the laser setup, the system will have to produce a synchronized external trigger output signal for driving a laser source through a pulse generator. This output trigger signal will drive the input of a standard pulse generator and the

frequency will be fixed to 1 kHz. In addition, the system will be able to delay this output signal for a synchronized acquisition with the laser setup.

- Fourth, the system must be connected to a host computer by using *universal serial bus* (USB). This computer will process and will store the data acquired. An USB connection will be used due to its good characteristics attending to its high transmission data rate, extensive use and plug-and-play capability, and it is widely available and accessible to at most every computer in the market.

- Fifth, the system will be controlled from a host computer by a software application in communication with a FPGA (*field programable gate array*) which will interpret and will execute the orders. A FPGA is preferred to a microcontroller due to its capability to integrate custom logic and embedded processing. This fact helps to minimize the hardware required for a specific digital purpose. In this sense, when fast digital logic circuits must be designed, where time delays are crucial, and non-discrete logic should be used, often a PLD (*programmable logic device*) is used instead. In this case, since some type of processing will be required for sequential operation as managing at high level the system, and also fast digital logic must be used for implementing low level tasks, the FPGA solution is optimal although its internal structure is different from a PLD. A more detailed explanation about FPGAs and their main characteristics will be carried out in the fifth chapter, fundamentally focused on the type used in this system. The computer software will control the whole system by means of the FPGA. This software will also process the data further. The idea is to take advantage of the computer processing power for the hard processing required for the data and leave the FPGA free for the low level tasks. The software will be designed for the Linux OS.

- Sixth, the system will have its own power supply system. The system will use a desktop voltage source as input voltage supply. From this main *DC* supply input, the voltage levels required will be produced in the system. The detector bias voltage will be generated by an external voltage source, since the current must be monitored and the voltage could be considerable depending on the detector used (more than 100 V in some cases and up to 1000 V for irradiated devices).

The main goal is to process and digitalize the analogue signal coming from the silicon detector channels as well as to manage all the logic required for the different types of setups. The analogue front-end electronics that incorporates the

Beetle chip is composed of a charge sensitive amplifier and a pulse shaper of first order (these electronic stages were described in the second chapter). Therefore, a voltage pulse signal is generated from the detector current pulse. This voltage pulse is the signal that is digitalized.

3.3. System architecture

The system has been divided into two main parts, a hardware part and a software part (figure 3.5), following the system requirements stated in section 3.2. The hardware part has to acquire the microstrip silicon sensor signals either from an external trigger input, in case of radioactive source setup is used, or from a synchronized trigger output generated by the system, if a laser setup is used. These acquired data are roughly processed and sent by the hardware in order to be stored in a host computer for a more sophisticated processing.

The hardware part is a dual board based system composed by a mother board and a daughter board. The mother board is intended to process the analogue data that comes from the readout chips, to manage the trigger signals, to control the whole system and to communicate with a host computer via USB. The daughter board is a small board intended to contain two Beetle readout chips, pitch-adaptors and detector support to interface the sensors. Hence, most of the hardware required must be implemented in the mother board. Particularly, the following hardware blocks should be implemented in the mother board: the trigger output generation stage for the laser setup, the trigger input processing stage for the radioactive source setup, the signal conditioning and digitalization of the analogue data coming from the Beetle chips, the control and configuration generation block for the Beetle chips, the USB communication controller, an external memory for temporary storage of the acquired data, the FPGA and associated circuits and the power supply generation block for the required voltage levels.

The daughter board contains the hardware required for accommodating two Beetle readout chips, for buffering the analogue data sent to the mother board, for receiving the control and configuration signals for the Beetle chips, for sending to the mother board a temperature signal, for connecting the microstrip silicon sensor(s) to the beetle chips and for biasing the detector(s) from an external voltage source.

The main reason for dividing the hardware into two boards is to prevent the rest of the hardware from the aggressive environment (radiation or very low temperatures) that will suffer the detectors. Both boards are communicated by flat ribbon cable for the analogue data signals coming from the Beetle chips, slow and

fast control digital signals to the Beetle chips, a thermistor signal as well as the supply level for the Beetle chips and buffers. The high voltage detector power supply is provided directly to the daughter board. Therefore, the daughter board is intended to be placed close to the radiation source whereas the mother board can be near the host computer. The length of the flat cable can be about various metres without any need of signal equalization.

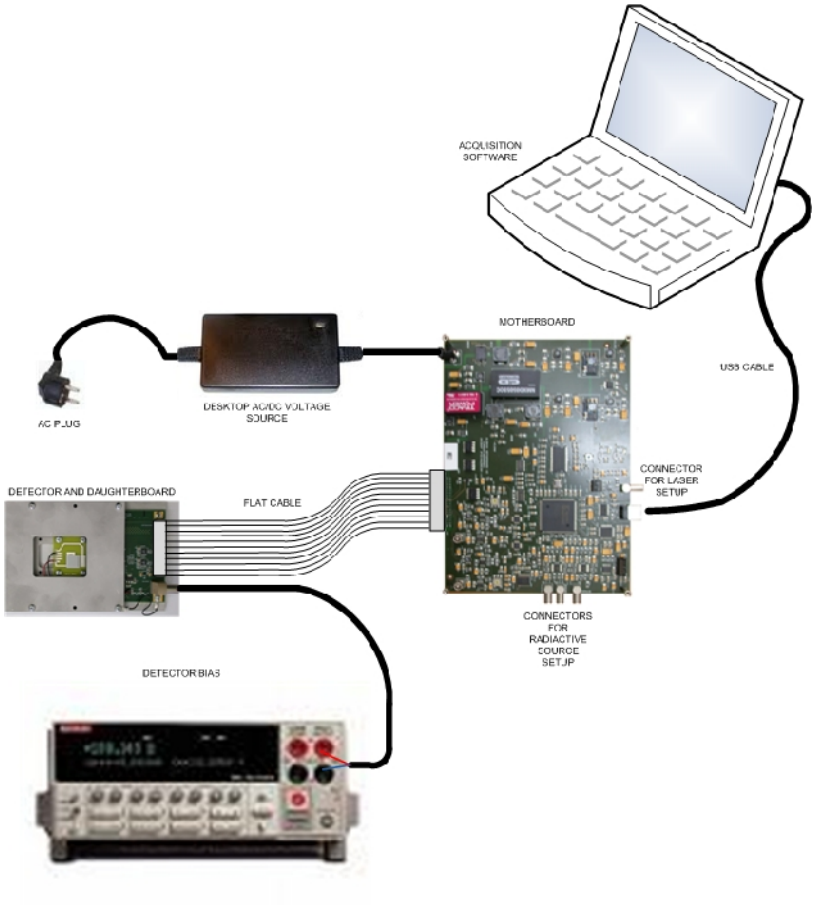

Figure 3.5. *Block diagram of the system with its different components.*

Regarding the software part of the system, its function is controlling the whole system and processing the data acquired from the sensors in order to store it in an adequate format file. The software must control the whole system for configuration, calibration and acquisition. Second, the software has to be the interface between the system and the user by means of a g*raphical user interface*

(GUI). Third, it must generate information about the acquisition and store it in adequate output file format. This format file will be compatible with software used for further data analysis. Finally, the software should monitor the state of the system in different situations.

The software has been designed using two levels. A low level for communication between the software and the mother board by USB has been implemented. No driver design is required at this level since the manufacturer of the USB controller provides with a ready-to-use driver (*virtual com port driver*). A high level for GUI, system monitoring and output file generation has been designed as well. The software is compatible with Linux. With this software the system will be able to be configured and calibrated. Acquisitions with a laser setup or a radioactive source setup will be able to be carried out as well.

Chapter 4

The daughter board

The design of the daughter board is treated in this chapter. In section 4.1, the block diagram of the daughter board is explained in order to have an overall view of the board hardware. Then the different blocks of the block diagram are described. The Beetle chip operation and its configuration for this design are discussed in section 4.2. In section 4.3, the design of the analogue buffers for the analogue output signal of each Beetle chip is detailed. The fast control and slow control schemes followed in this system are explained in section 4.4. In section 4.5, the power supply system used for biasing the daughter board Beetle chips, analogue buffers and detectors is considered. The fan-ins used to connect the Beetle chips and the detector(s) are described in section 4.6. Finally, the daughter board PCB design, the detector(s) PCB design and the test box design are considered in section 4.7.

4.1. Daughter board block diagram

As it has been explained in chapter 3, the daughter board is a small board equipped with the minimum components needed for accommodating two Beetle chips [41], a temperature sensor, the board supply regulation as well as the microstrip silicon sensors support, that is, pitch adaptors for connecting the detectors to the Beetle chips and the detector bias stage.

The block diagram of the daughter board is shown in figure 4.1. There are two *Beetle* readout chips used to read out up to 256 channels of microstrip detector(s). The analogue outputs of both chips are buffered and sent to the mother board in

parallel. The slow control signals, using an *inter-integrated circuit* (I2C) bus [42], and the fast control signals, with *low voltage differential signaling* (LVDS) standard [43], required for controlling and configuring the chips are sent from the mother board and shared by both chips.

There is a NTC (*negative temperature coefficient*) thermistor as close as possible to the detectors in order to have a temperature readout on each acquired event as well. A *dc* supply level (5V) is sent from the mother board. This supply level is regulated by three *low drop out* (LDO) linear regulators for obtaining the *dc* supply levels required by the readout chips (2.5 V) and the buffer stage (3 V).

The connection between the daughter board and the mother board is carried out by means of IDC (*insulated displacement connector*) connectors and flat ribbon cable. The detector high voltage reverse bias is supplied directly to the daughter board through a *Lemo* power connector. There is also a RC decoupling stage for this supply prior to the detector for minimizing the noise in the detector bias. The different blocks of this block diagram will be explained in the following sections.

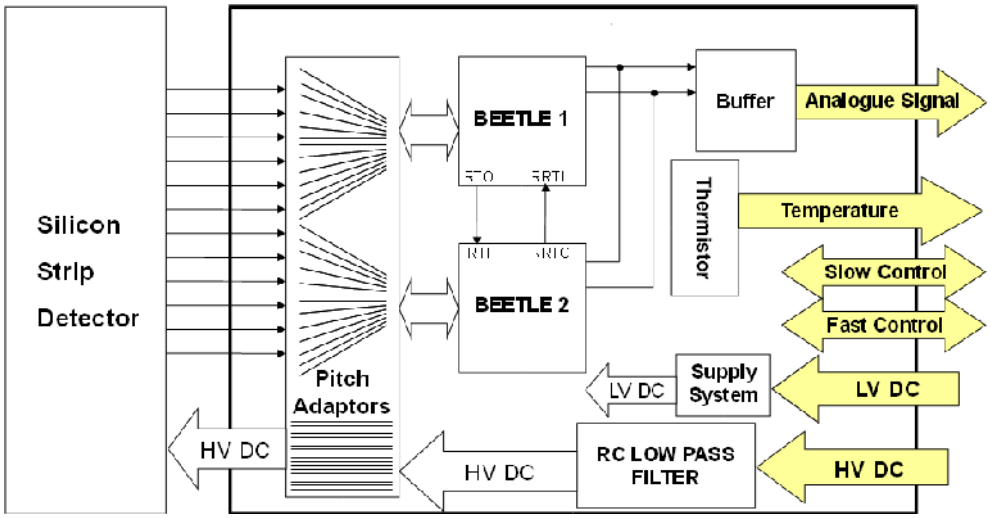

Figure 4.1. *Block diagram of the daughter board.*

4.2. The Beetle chip

The Beetle is a readout chip designed for the LHCb experiment and it is used in the Vertex detector, the Silicon tracker detector and the RICH detector. The Beetle chip has been developed at the ASIC laboratory of the University of Heidelberg. The main characteristics of the Beetle chip can be found on [41, 44, 45]. The

schematic block diagram of the Beetle readout chip is shown in the figure 4.2. The Beetle can be operated either as analogue or as binary pipelined readout chip.

4.2.1. Chip architecture

The chip is composed of 128 channels. Each channel consists of a low-noise charge-sensitive preamplifier, an active CR-RC pulse shaper and a buffer, forming together the analogue front-end. A test channel and a sense channel are included as well as the 128 front-end channels. The shape of the front-end pulse can be tailored to the specific requirements of the application. The input signal coming from the detector can be either a positive or a negative current pulse. Each channel has a comparator, which can be configured to discriminate the front-end output pulses. The threshold is adjustable for each channel and input signals of both polarities can be processed by the comparator.

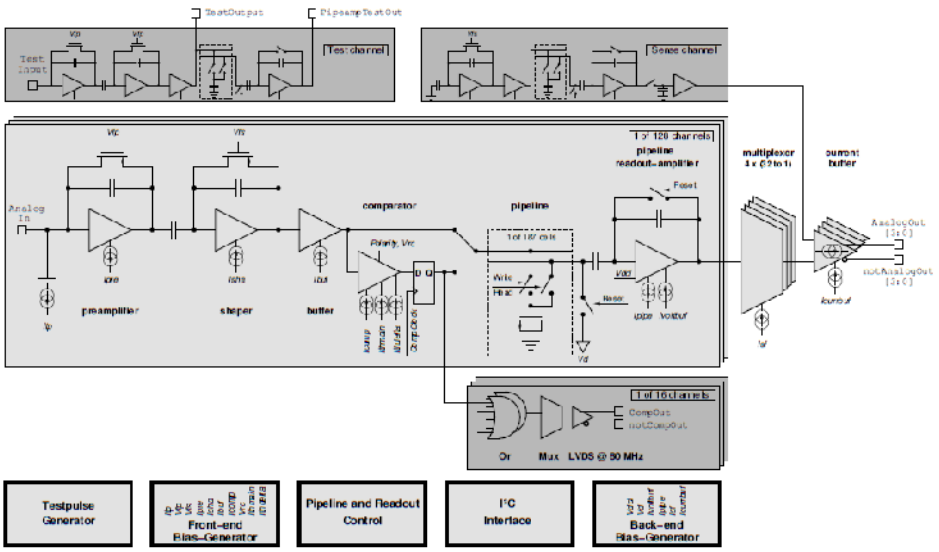

Figure 4.2. *Block diagram of the Beetle readout chip. Figure taken from [41].*

Either the front-end output or the comparator output is sampled with the clock frequency (usually 40MHz) into an analogue pipeline memory. The pipeline is a switched-capacitor array of 130 × 187 cells, with a programmable latency of maximum 160 sampling intervals and an integrated multi-event buffer of 16 stages. The signal stored in the pipeline is transferred to the multiplexer via a resettable charge-sensitive amplifier. Current drivers bring the serialised data off chip either using one port or four ports for a lower readout time. The output of the sense

channel is subtracted from the analogue data (the other 128 channels) to compensate common mode effects. The chip also provides a prompt binary readout mode, independently of the aforementioned pipelined modes, in which the discriminator output of the comparator is synchronized with an externally provided comparator clock. Four adjacent channels are logically ORed and routed directly off chip via LVDS drivers at double data rate.

All amplifier stages are biased by means of on-chip DACs with 8 bit resolution which generate the bias currents and voltages. For test and calibration purposes a charge injector with adjustable pulse height is implemented on each channel. The bias settings and various other parameters like the trigger latency can be controlled via a standard I2C-interface. All digital control and data signals, except those for the I2C ports, are routed via LVDS ports.

The chip has been implemented in a 0.25 μm standard CMOS process technology. It has been designed to ensure the radiation hardness against total dose effects in excess of 130Mrad. The *Single Event Latch-up* (SEL) is suppressed by means of guard-rings. The continuous use of triple-redundant logic ensures robustness against *Single Event Upset* (SEU).

4.2.2. Analogue front-end operation

The analogue front-end is composed of a charge-sensitive preamplifier, a CR-RC shaper and a buffer. The principle of operation of the CSA and the CR-RC shaper has been explained in chapter 2. The analogue front-end output signal is shown in figure 4.3.

The front-end output signal is a semi-Gaussian pulse characterized by three parameters: the peaking time t_p (time from 0 to 100% of the pulse height), the peaking voltage V_p (maximum voltage of the pulse) and the remainder R (ratio between the voltage 25 ns after the peak voltage, V_{25+}, and the peak voltage).

The pulse shape can be varied by 5 bias parameters (preamplifier bias current, I_{pre}, shaper bias current, I_{sha}, buffer bias current, I_{buf}, preamplifier feedback resistance, V_{fp}, and shaper feedback resistance, V_{fs}) of the front-end electronics which can be configured by means of the I2C slow control. In addition, the pulse shape is strongly coupled to the input capacitance of the detector.

The peaking voltage (V_p) is proportional to the collected charge, Q, in the corresponding input channel. Under nominal bias settings (I_{pre} = 600 μA, I_{sha} = I_{buf} = 80 μA, V_{fp} = V_{fs} = 0 V), at an ambient temperature of 25°C and with an input

capacitance of 3 pF, the theoretical sensitivity of the front-end circuit, A_Q, is given by [41]

$$A_Q = \frac{V_p}{Q} = \frac{38\,mV}{22000\,e}. \quad (4.1)$$

The peak voltage (V_p), the peaking time (t_p) and the remainder change (V_{25+}) with the temperature and with the input capacitance (table 4.1). For design purposes, it is interesting to reconstruct the whole pulse shape, including the undershoot, so a maximum pulse length of 100 ns is considered.

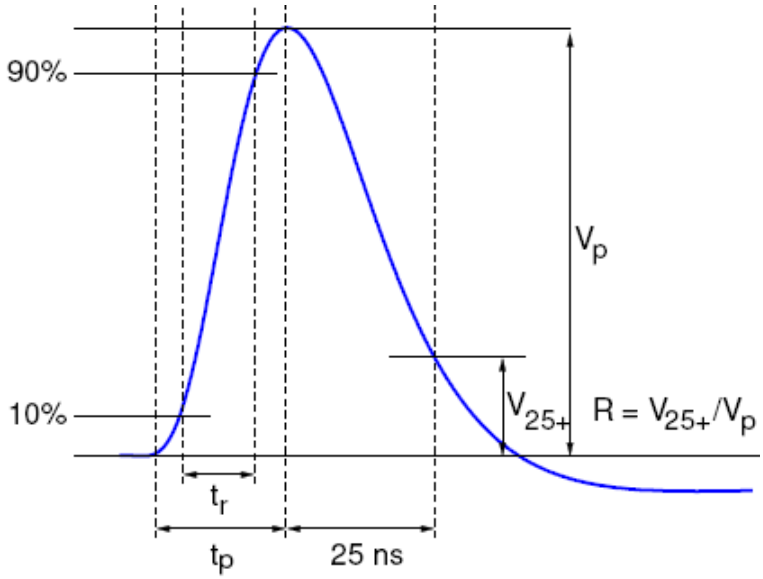

Figure 4.3. *Semi-Gaussian pulse shape output of the Beetle front-end with the corresponding parameters used to characterize the shape. Figure taken from [41].*

The nominal equivalent noise charge (ENC) of a complete Beetle (version 1.5) can be determined by the following expression [44],

$$ENC = (531 \pm 10)e + (49.8 \pm 0.5)\frac{e}{pF} \cdot C_i \quad (4.2)$$

where C_i is the input capacitance seen by the chip in each input channel (detector load capacitance as well as the parasitic capacitance). This expression is for nominal conditions and 25°C. The offset of the ENC expression will increase with the temperature almost linearly with a slope of about 0.8 e/K over a range from

[44].

C_i (pF)	Peaking ratio (V_p/V_p, $C_i = 3pF$)	Peaking time t_p (ns)	Remainder R (%)
3	1	21 ± 0.5	2.8 ± 1.2
7	0.87 ± 0.3	22.5 ± 0.5	5.8 ± 1.5
12.9	0.79 ± 0.3	23.5 ± 0.5	10.1 ± 1.7
25.7	0.67 ± 0.4	25.8 ± 0.6	23.9 ± 1.8
36.3	0.59 ± 0.4	27.7 ± 0.6	31.3 ± 2.0
50.5	0.51 ± 0.5	28.2 ± 0.7	33.8 ± 2.1
T(°C)	Peaking ratio (V_p/V_p, $T = 25°C$)	Peaking time t_p (ns)	Remainder R (%)
-40	1.45 ± 0.4	16.5 ± 0.4	-32 ± 2.5
-30	1.4 ± 0.4	17 ± 0.4	-28 ± 2.5
-15	1.3 ± 0.4	16.7 ± 0.4	-20 ± 2
0	1.29 ± 0.4	17.5 ± 0.5	-15 ± 2
15	1.18 ± 0.3	18 ± 0.5	-8 ± 1.5
30	0.96 ± 0.3	19 ± 0.5	1 ± 1.5
45	0.86 ± 0.3	20.5 ± 0.5	7 ± 1.5
60	0.77 ± 0.2	22 ± 0.5	10 ± 1.5

Table 4.1. *Measured pulse shape parameters for different input capacitances Ci (at an average ambient tempreature between 40°C and 45°C) and ambient temperatures T (at constant input capacitance of 3 pF). All front-end bias settings are programmed to the nominal values. Data extracted from [44].*

4.2.3. Pipeline operation and output format

The front-end analogue output without discrimination is used for this readout system since the output pulse of irradiated and non-irradiated silicon detectors will be acquired. The analogue output of each channel is then sampled into the analogue pipeline which works as a ring buffer. The analogue pipeline has 130 rows and 187 columns, *i.e.* 130 x 187 cells. Each cell can be considered as a capacitor where the sampled pulse voltage value is stored if a switch is on. The voltage switch is a MOS transistor [44, 45] in each cell. Therefore, the pipeline is implemented as a switched capacitor array. The sampling frequency is the same as the clock frequency used for operating the Beetle chip. For this system the clock frequency has been fixed to 40 MHz. Therefore the front-end output is sampled every 25 ns. The specific point of the pulse sampled will vary with time depending on the

moment when the detector signal is received, *i.e.*, when a charged particle has crossed the detector.

The latency of the pipeline can be defined as the temporal distance between two pointer signals which control the writing on each column and the readout of each column respectively. The latency of the pipeline can be configured by *slow control* (I^2C interface). In this application, the latency has been fixed to 128 clock cycles, corresponding to 3.2 μs. The writing of each pipeline column takes place every 25 ns. The readout of each pipeline column is controlled by the *Trigger fast control* signal (LVDS interface). Hence, the *Trigger* signal will have to be active 128 clock cycles, *i.e.* the latency, after a particular front-end pulse point of interest has been sampled and written in the analogue pipeline.

The analogue readout of the Beetle chip can be carried out either onto one port or onto four ports. The readout onto four ports is faster. The analogue readouts of each Beetle chip have been configured as single readouts onto one output port. Since the event rate of the laser setup will be 1 kHz and the event rate of the radioactive source setup is slower than the laser setup one (actually the radioactive source setup rate event is random but slower in general), the readout onto one port has been chosen in order to minimize the signal conditioning hardware. Therefore, the system has two analogue output ports, one from each Beetle chip. Single readouts will be acquired because of the low setups rate event. A consecutive readout would take place if a second trigger occurs before the last readout is completed.

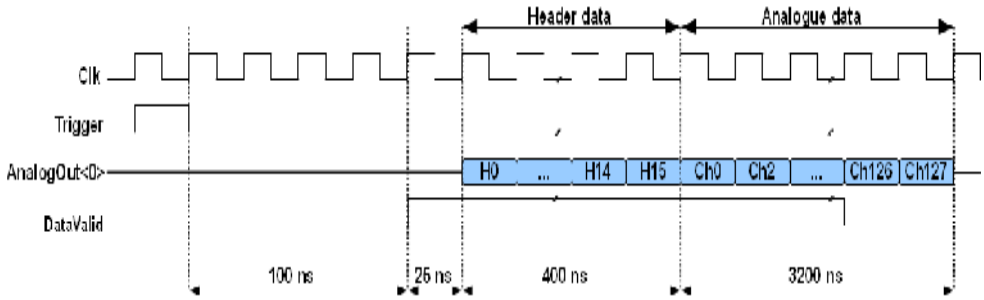

Figure 4.4. *Readout timing scheme of the analogue single readout mode onto one port for the Beetle chip.*

The analogue output format for a single readout onto one port is shown in figure 4.4. The readout is a multiplexed signal with a 16 bits header and the 128 multiplexed channels (the sampled voltage value for each channel). The width of each header or each channel depends on the frequency of the readout clock. The frequency of this clock can be configured by slow control and it is usually the same

as the sampling clock. For this application, the frequency of both clocks is the same and it is fixed to 40 MHz. A readout takes place 150 ns after the *Trigger* signal has been activated. The *Trigger* signal is sampled internally to the negative edge of *Clk*. A *DataValid* signal is used to inform that a readout is going to occur imminently. This *DataValid* signal is activated 25 ns before the first header bit is readout and it will continue active for 16 header bits and 126 channels (3575 ns). The length of a single readout onto one port is 3600 ns. The readout is synchronous to the clock signal used. The header bits give information about the specific readout, like the pipeline column number which has been read. In the figure 4.5 a real analogue readout onto one port is shown.

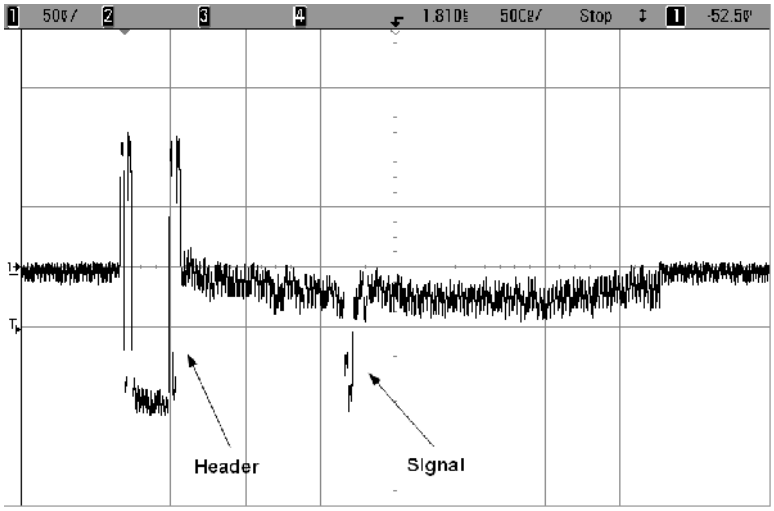

Figure 4.5. *Analogue single readout onto one port for the Beetle.*

4.2.4. Calibration

For calibration purposes, internal test pulses can be generated by the Beetle chip from a test pulse generator coupled to the preamplifier on each input channel. A step-like pattern corresponding to +1 and −1 times the reference signal amplitude is coupled to the input channels. Its amplitude alternates with the channel number and can be adjusted with the I_{tp} bias register (table 4.2) by slow control configuration. From one test pulse injection to the next, the polarity of the charge injection is changed in each channel. The test pulse generators are controlled by the fast control *Testpulse* signal. A test pulse is triggered via the rising edge of this signal and can be enabled per channel by means of the slow-control configurable *TpSelect* register (table 4.2). The test pulse is independent from the Beetle clock signal. The relation between I_{tp} and the injected charge Q_{in} is given by [41, 44]

$$Q_i = 1025\, e \cdot I_{tp}[regbit] \quad (4.3)$$

where *Itp[regbit]* is the content of the configuration register *Itp* (from 0 to 255). The timing relation between *Testpulse* and *Trigger* is depicted in figure 4.6.

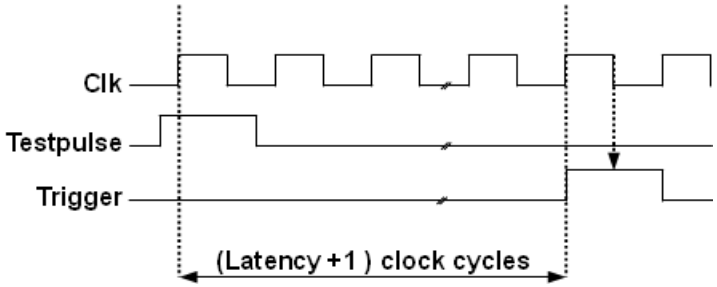

Figure 4.6. *Timing relation between Testpulse and Trigger. Latency refers to the content of the latency register.*

4.2.5. Configuration registers and Slow control

The Beetle contains 24 addressable registers. Registers 0-15 are bias registers for the analogue stages. Register 16 defines the chip's latency, which must be between 10 and 160 for reliable chip operation. Registers 17, 18 and 19 select the mode of operation. Registers 20, 21 and 22 are operated as shift-registers, *CompMask* and *TpSelect* are 128-bit register each, segmented in 16 8-bit registers, and *CompChTh* establishes a 1024 (128 × 8) bit register.

The Beetle slow control interface is a standard mode I²C-slave device with a transfer rate of 100 kbit/s. The chip address, which is necessary to access a single device via the I²C-bus, is 7 bits wide and it is assigned via the address pads I2CAddr[6:0]. The internal configuration registers are accessed by means of a pointer register containing the address of the register to be written or read first. The pointer is internally incremented by one after each transferred data frame so that registers with adjacent addresses can be accessed consecutively. The pointer register itself remains unchanged, that is, a new transfer will start at the original pointer position. The registers with addresses 20−23 cannot be accessed consecutively since registers 20−22 are implemented as 128-bit shift registers and register 23 is the output of the SEU counter. In the table 4.2 are listed the Beetle configuration registers, a more detailed description of these registers can be found in [41].

Reg. n°	Reg.Name	Range	Res. LSB	Nominal setting		Description
				Value	*Reg. content*	
0	I_{tp}	0 - 2 mA	7.8 µA	0 µA	0x00	test pulse bias current
1	I_{pre}	0 - 2 mA	7.8 µA	600 µA	0x4C	preamplifier bias current
2	I_{sha}	0 - 2 mA	7.8 µA	80 µA	0x0A	shaper bias current
3	I_{buf}	0 - 2 mA	7.8 µA	80 µA	0x0A	front-end buffer bias current
4	V_{fp}	0 - 2.5 V	9.8 mV	0 mV	0x00	preamplifier feedback voltage
5	V_{fs}	0 - 2.5 V	9.8 mV	0 mV	0x00	shaper feedback voltage
6	I_{comp}	0 - 2 mA	7.8 µA	40 µA	0x05	comparator bias current
7	$I_{thdelta}$	0 - 2 mA	7.8 µA	—	—	current defining incremental comparator threshold
8	I_{thmain}	0 - 2 mA	7.8 µA	—	—	current defining common comparator threshold
9	V_{rc}	0 - 1.25 V	4.9 mV	0 mV	0x00	comparator RC constant
10	I_{pipe}	0 - 2 mA	7.8 µA	100 µA	0x0D	pipeamp bias current
11	V_d	0 - 2.5 V	9.8 mV	1275 mV	0x82	pipeamp reset potential
12	V_{dcl}	0 - 2.5 V	9.8 mV	1030 mV	0x69	pipeamp reference voltage
13	$I_{voltbuf}$	0 - 2 mA	7.8 µA	160 µA	0x14	pipeamp buffer bias current
14	I_{sf}	0 - 2 mA	7.8 µA	200 µA	0x1A	multiplexer buffer bias current
15	$I_{currbuf}$	0 - 2 mA	7.8 µA	800 µA	0x66	output buffer bias current
16	*Latency*	10 - 160	—	160	0xA0	trigger latency
17	*ROCtrl*	—	—	—	—	readout control
18	*RclkDiv*	0 - 255	—	0	0x00	ratio between *Rclk* and *Sclk*
19	*CompCtrl*	—	—	—	—	comparator control
20	*CompChTh*	0 - 31	—	—	—	comparator channel threshold shif register implementation
21	*CompMask*	—	—	0	0x00	comparator mask shift register implementation
22	*TpSelect*	—	—	0	0x00	test pulse selection shift register implementation
23	*SEUcounts*	0 - 255	—	—	—	sum of Single Event Upsets

Table 4.2. *Beetle configuration registers. Data extracted from [41].*

4.2.6. Reset modes

The chip has a power-up reset and an external reset. The power-up reset is activated immediately when the power of the chip is switched on. The time until the power-up reset becomes inactive is adjusted via an external capacitance connected to the *PowerupReset* pad. For the capacitance value used of 100 nF the

time constant results in 280 ms. All Beetle registers are reset to zero and the I2C-interface is initialised.

The external reset is controlled via the *fast control Reset* signal. It resets the pipeline write and trigger pointer to column number zero and initialises the control logic state machines. The rising edge of this external reset initialises also the I2C-interface. The minimum reset width is one sampling clock cycle. The external *Reset* signal is sampled internally to the negative edge of *Clk*.

4.3. Analogue buffers

The analogue output signals of the Beetle chip are current differential signals since the output driver is a fully differential *operational transconductance amplifier* (OTA) [44-46]. The loop for each signal is closed with a 100 Ω resistor on the daughter board and the voltage over that resistor is the analogue output voltage. In table 4.3 are shown the current output driver levels measured over a 100 Ω resistor. The voltage difference is obtained from the chip pads *notAnalogOut[0]* minus *AnalogOut[0]* across the resistance for each Beetle chip. The analogue levels for the 128 analogue channels will correspond to the front-end pulse sampled value.

The Beetle analogue outputs must be buffered at the daughter board since they have to be sent to the mother board via a ribbon flat cable. In order to have an estimation of the maximum signal bandwidth, a square signal with the header levels of table 4.3, a period of 50 ns, a pulse width of 25 ns with rise and fall times of 4 ns, was considered (figure 4.7). The rise and fall times of the signal have been estimated conservatively taking into account the output multiplexer design and the analogue output driver of the Beetle chips [44-46].

		$V_{AnalogOut}$ (mV)	$V_{notAnalogOut}$ (mV)	I_{out} (mA)
Baseline		973	978	-0.05
Header	**High**	916	1058	-1.42
	Low	1014	912	1.29
Data	**High**	analogue readout		
	Low			

Table 4.3. *Beetle current output driver levels measured over a 100 Ω resistor. Data collected from [41].*

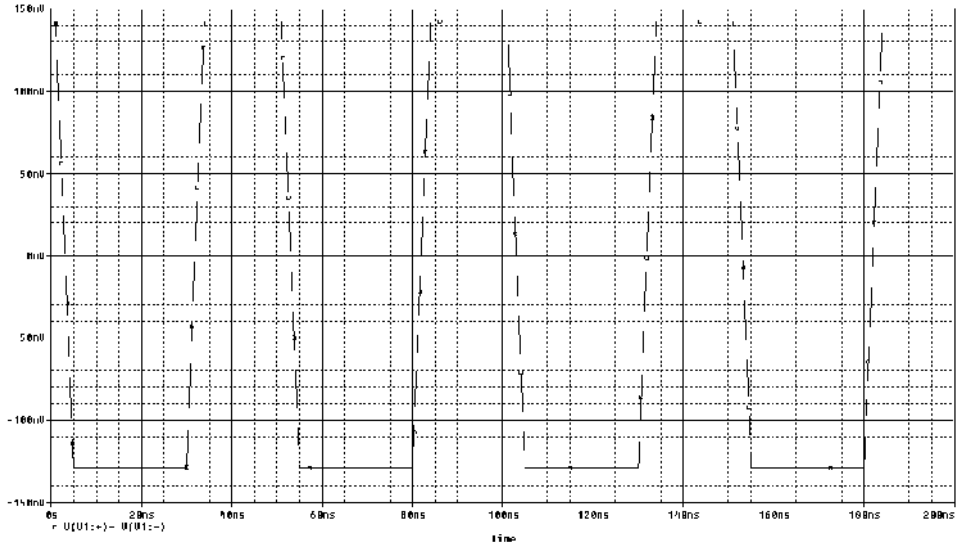

Figure 4.7. *Beetle analogue output signal approximation for estimating its maximum bandwidth. On the x axis the time is represented in 20 ns intervals. On the y axis the voltage is represented in 50 mV intervals.*

The frequency content of interest from the fall time or the rise time (the lower of these times) can be estimated with the following relation [47]

$$f_{knee} = \frac{1}{2 \cdot t} \quad (4.4)$$

where t is the lower time between the rise time and the fall time and f_{knee} is the frequency where the frequency spectrum of the signal is 6.8 dB below the -20 dB/dec slope of the Fourier envelope. The spectrum amplitude of the signal for frequencies higher than f_{knee} decreases faster (with a -40 dB/dec slope) and its influence on the signal behaviour is very limited. For the signal values considered f_{knee} is 125 MHz, so an analogue signal bandwidth of 125 MHz has been estimated for the design. In the figure 4.8, a *Fast Fourier Transform* (FFT) representation of the signal shown in figure 4.7 can be seen. The spectrum content of the figure 4.8 is in agreement with the estimated bandwidth of 125 MHZ for a rise time and a fall time of 4 ns.

For better common-mode and power supply noise immunity, a differential signalling mode would be optimum for these Beetle chips analogue signals [47-49] since these signals will have low voltage levels and high bandwidth (up to 125 MHz). Differential transmission uses two wires with opposite current-voltage

swings to convey data information instead of the one wire used in single-ended methods. The advantage of the differential approach is that noise is coupled to both wires as common mode, *i.e.* the noise appears on both lines equally, and is thus rejected by receivers sensitive only to the difference between both signals. Differential signals also radiate less noise than single-ended signals due to the cancellation of magnetic fields.

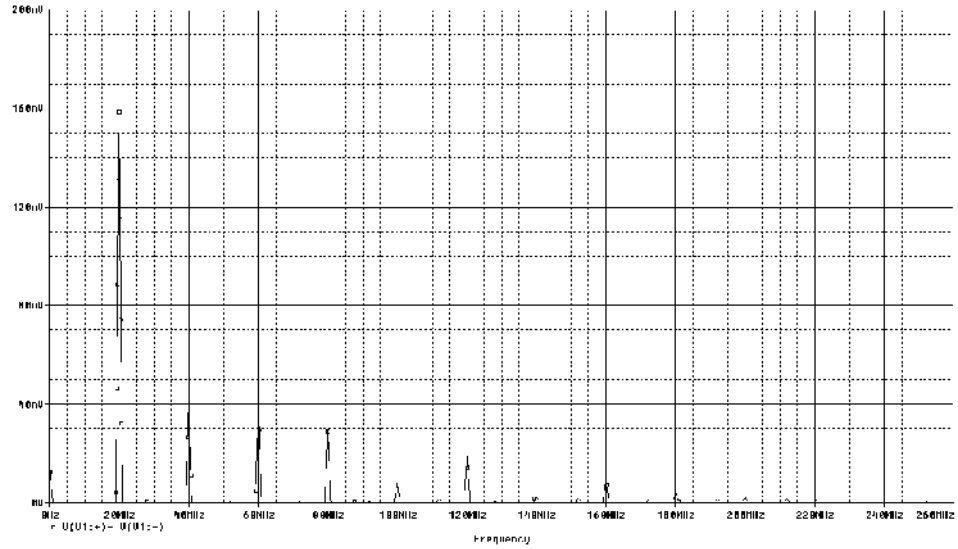

Figure 4.8. *Fast Fourier Transform representation for the signal of the figure 4.7. On the x axis the frequency in 20 MHz intervals is represented. On the y axis the voltage in 40 mV intervals is represented.*

Taking into account these considerations, a differential line driver (AD8132 from *Analog Devices*) has been used in order to buffer the analogue output signals. The AD8132 [50] is a high speed differential amplifier. The schematic circuit of one of the two buffer stages of the daughter board is shown in the figure 4.9. The buffer has an overall unity gain. However, the differential amplifier gain has been fixed to a factor of two with a bandwidth of 160 MHz for this gain. The differential gain of the amplifier is controlled by means of the resistor rate given by

$$G_{diff} = \frac{R_4}{R_2} = \frac{R_5}{R_3} \quad (4.5)$$

where *Gdiff* is the differential gain. The common mode gain related to the differential inputs is zero by design (in fact, it is dominated by the *common mode rejection ratio*, CMRR). The output common mode is adjusted by means of the

common mode voltage input. The common mode gain related to this common mode input is one. The common mode of the output signal has been fixed to 1.5 V (mid-range at the supply levels) by means of a voltage divider. The amplifier differential output signal must be impedance matched in order to maintain the signal integrity avoiding reflections and other problems. For this purpose, two resistors of 51 Ω (*R6* and *R7*) have been included for each line. These resistors will match the line impedance (flat ribbon cable) in conjunction with a 100 Ω resistor at the end of the line in the mother board, avoiding signal reflections. This impedance matching scheme results on halving the signal dynamic range at the end of the line, having an overall gain for the buffer of one.

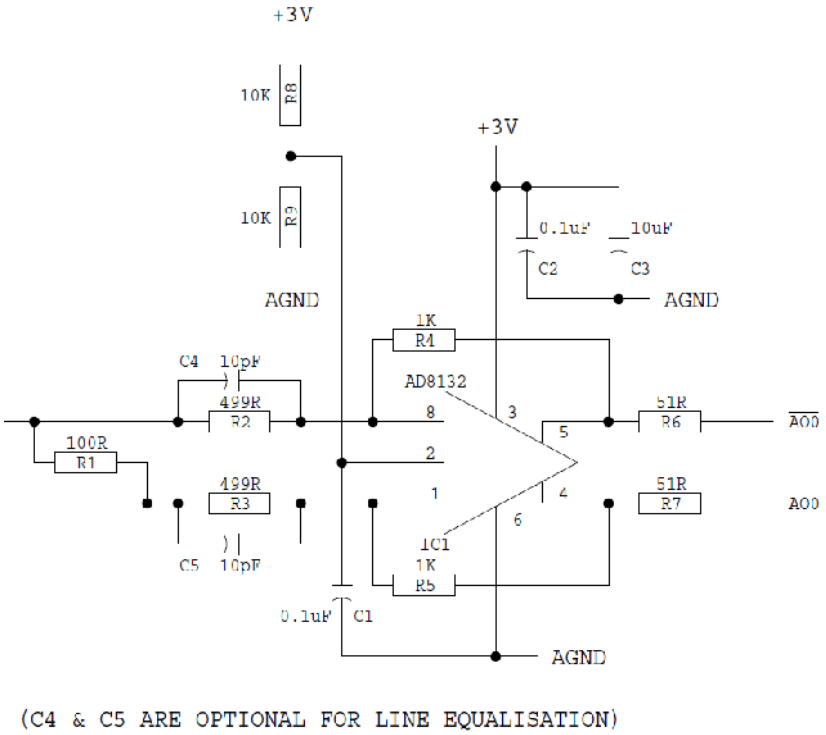

Figure 4.9. *Schematic diagram of one buffer circuit of the daughter board.*

The supply levels for the differential amplifiers are 3V and 0V. Decoupling capacitors (*C2* and *C3*) with values recommended by the manufacturer have been placed for minimizing the power supply noise. Considering the power supply levels and the output common mode voltage, saturation effects on the amplifier will be present for output signals higher than about ± 500 mV over 1.5 V. The Beetle analogue output dynamic range is shown in the figure 4.10. With the current

buffer gain configuration (unity gain), the dynamic range of the analogue outputs from the daughter board would cover the linear region of the graph without saturation effects (approximately ±110000 e-). As a first approximation, it would be sufficient for the system. The overall system gain would be able to be adjusted later in the system development.

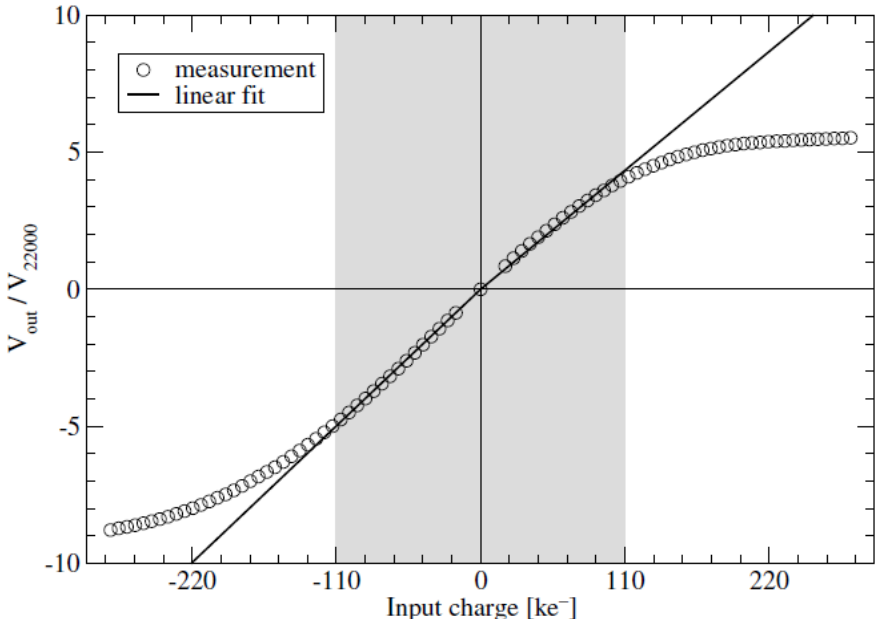

Figure 4.10. *Dynamic range of the Beetle chip for positive and negative input charges. At the ordinate the analogue output signal Vout is normalised to V_{22000}, which is the corresponding voltage for an input charge of 22 000 e (38 mV for nominal bias parameters and 25 °C). The highlighted area between ±110000 e shows the linear region with a maximum variation of ±5%. Figure taken from [44].*

Concerning the length of the cable between the daughter board and the mother board, no line equalization is required at the daughter board for the analogue signals if shielded twisted pair cable no longer than 10 m would be used [51]. This length will be reduced to 5 m since ribbon flat cable is used to connect the daughter board and the mother board, and both the frequency response and the crosstalk deteriorates with respect to the shielded twisted pair cable [47]. However, this flat ribbon cable can be twisted (*twist and flat* ribbon cable) for the analogue and digital differential signals if required, improving both the frequency response and the crosstalk attenuation. On the other hand, the signal attenuation due to the length of the transmission line is worse at higher frequencies than at lower frequencies. One way to compensate for this is to provide an equalizer circuit that boosts the higher frequencies in the transmitter circuit, so that the attenuation effects are

diminished at the end of the cable. Therefore, two capacitor footprints (C_4 and C_5) have been placed in the daughter board PCB if line equalization is required, as can be seen in figure 4.9.

4.4. *Fast control* and *Slow Control*

There are four *fast control* signals for operating the Beetle chip: *Clk*, *Reset, Testpulse* and *Trigger*. These signals are generated by the FPGA at the mother board and sent to the daughter board by means of flat ribbon cable. There are also two *DataValid* signals, one for each Beetle chip, which are generated by the Beetle chips when a readout is going to take place and sent to the mother board via the flat ribbon cable.

The aforementioned six signals follow LVDS standard [43][52]. The LVDS standard is a way to communicate data using a very low voltage swing that fits the requirements for these *fast control* signals. This format allows single channel data transmission at hundreds of megabits per second (Mbps). It has low swing and current mode driver outputs which create low noise and provide low power consumption across frequency. The driver output consists of a current source (3.5 mA nominal) which drives one of the differential pair lines (see figure 4.11). The receiver has high *dc* impedance, it does not source or sink *dc* current, so the majority of driver current flows across the 100Ω termination resistor, generating about 350 mV across the receiver inputs. When the driver switches, it changes the direction of current flow across the resistor, thereby creating a valid one or zero logic state.

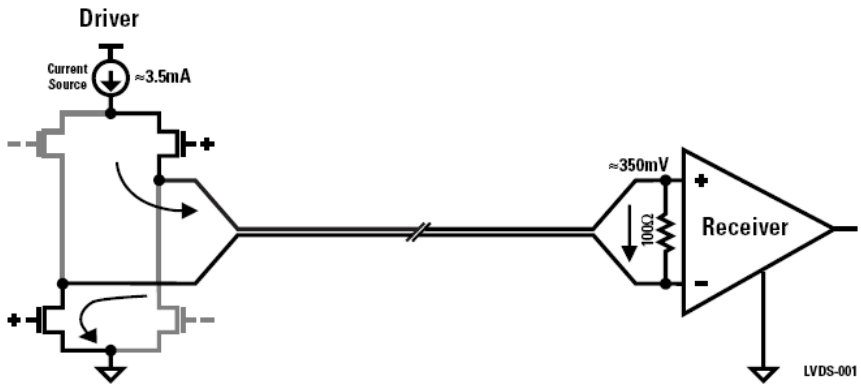

Figure 4.11. *Simplified diagram of a LVDS driver and a receiver connected via 100 Ω differential impedance media. Figure taken from [43].*

The fast control signals received by the daughter board (*Clk*, *Testpulse*, *Reset* and *Trigger*) are terminated each one with 100Ω termination resistor before they are connected to the Beetle chip pads. These signals are shared by both chips, *i.e.* each signal is split into two traces in the daughter board after the IDC connector in order to be connected to the corresponding Beetle chip input pads. The *DataValid* signals are sent directly to the mother board, where each one is terminated with a 100Ω resistor.

The slow control signals are the serial data line (SDA) and the serial clock line (SCL) of the I²C bus [42]. These signals are shared by both Beetle chips as well. Each device connected to the bus is software addressable by a unique address and simple master/slave relationships exist at all times; masters can operate as master-transmitters or as master-receivers. In this design the master transmitter/receiver is an I²C controller in the FPGA at the mother board and the Beetle chips are operated as slaves. Each Beetle chip has its own seven bits address: Beetle 0 has the '0100000' address and the Beetle 1 has the '0100001' address. Serial, 8-bit oriented, bidirectional data transfers can be made at up to 100 kbit/s in the standard mode.

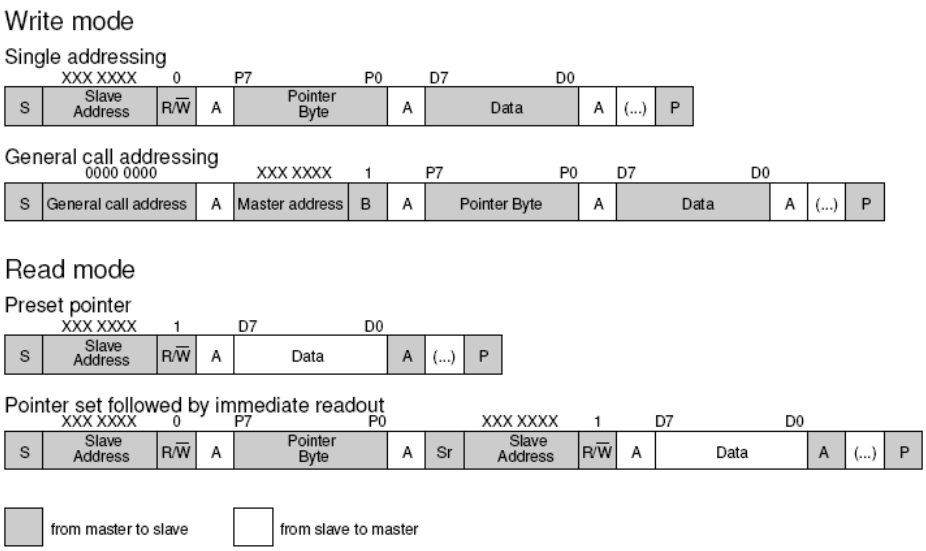

Figure 4.12. *I²C-bus write and read sequences for accessing registers on the Beetle. Figure taken from [38].*

Figure 4.12 explains the transfer sequences in write and read mode for the Beetle chip. Data is always transferred with the most significant bit (MSB) first. In write

mode the chip address is transmitted after transfer initialization, then the pointer byte and finally the data. After the transmission of one data frame, the pointer address is automatically incremented and it will address the following register. The transfer of the pointer byte is obligatory in write mode. In read mode there are two versions: preset pointer and pointer set followed by immediate read-out. In preset pointer, data are immediately read out after initialising the transfer and sending the chip address. The pointer has been set in a previous transfer. When a pointer is set and then it is followed by an immediate readout, the pointer byte is transferred after initialising the transfer and sending the chip address. Then the I2C-bus is initialised again, the chip address is sent and data is read out. The single addressing write mode and the pointer set followed by immediate readout are used in this system.

The number of ICs (integrated circuits) that can be connected to the same bus is limited only by a maximum bus capacitance. Both SDA and SCL are bidirectional lines, connected to a positive supply voltage, V_{DD} via a current-source or pull-up resistor. When the bus is free, both lines are *high*. The output stages of devices connected to the bus must have an open-drain or open-collector to perform the wired-AND function. Due to the variety of different technology devices (CMOS, NMOS, bipolar) that can be connected to the I2C-bus, the levels of the logical '0' (*low*) and '1' (*high*) are not fixed and depend on the associated level of V_{DD}. In this design the pull-up resistors are placed at the mother board. The V_{DD} level has been fixed to 3.3 V at the mother board as well. This level is compatible with the Beetle chips and the FPGA at the mother board.

4.5. Power supply system

The power supply system of the daughter board can be divided in the detector(s) bias system and the hardware supply system. Figure 4.13 shows the schematic diagram of the hardware supply system. The required levels for the daughter board hardware are the Beetle analogue supply (2.5 V), the Beetle digital supply (2.5 V) and the analogue buffer supply (3 V). These levels are generated by means of linear regulators from a floating power supply level generated at the mother board (5 V, 500 mA). This floating power supply level is sent to the daughter board via the same flat ribbon cable used for the rest of the signals.

The linear regulator used for generating the Beetle supply levels is the LM117-2V5 from *National Semiconductor* [53]. It is a low dropout (LDO) linear regulator with a dropout of 1.2V at 800mA of load current. A voltage regulator provides constant *dc* output voltage and contains circuitry that continuously holds the output voltage at the design value regardless of changes in load current or input voltage. The dropout voltage is defined as the minimum voltage drop required across the

regulator to maintain output voltage regulation. A linear regulator dissipates the least internal power and has the highest efficiency when operates with the smallest voltage across it. The LDO requires the least voltage across it. The capacitors are placed for the regulator stability and for noise minimization, and they have the recommended values from the manufacturer.

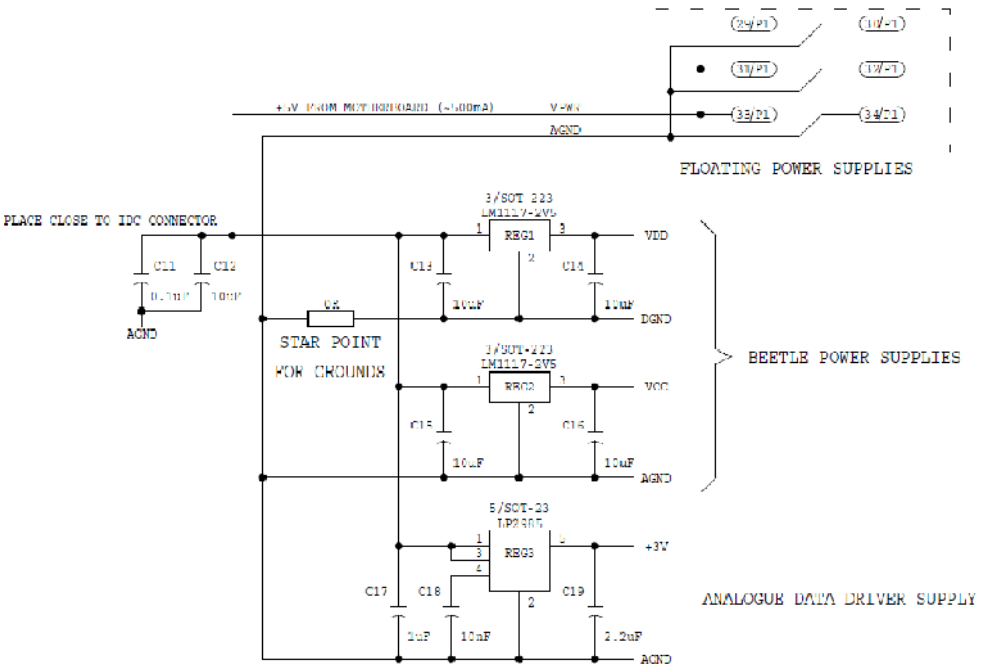

Figure 4.13. *Schematic diagram of the daughterboard hardware power supply system.*

The linear regulator for generating the analogue buffer supply is the LP2985 of *Texas Instruments* [54]. It is also a LDO linear regulator with a typical dropout of 280 mV at 150 mA load current. The capacitors have the recommended values from the manufacturer as well. As can be seen in the figure 4.13 the input power supply that comes from the mother board is filtered by to capacitors (C_{11} and C_{12}) prior to the input of the regulators in order to minimize the input noise coupled in the cable. Moreover, there is a star point connection for the different grounds for separating the analogue ground from the more noisy digital ground.

Regarding the detector(s) bias system, a schematic diagram of this supply system is shown in figure 4.14. The detector bias voltage is applied by means of an external power supply since the voltage applied can be high (up ±1000 V for irradiated detectors) compared to standard voltages in electronic systems.

Moreover, the current supplied by this source must be monitored in order to control the detector(s) leakage current.

The external detector bias voltage is applied trough a *Lemo* power connector. Prior to be supplied to the detector(s), this voltage is decoupled by means of two RC low pass filter stages. The cut-off frequency (-3 dB frequency) of these passive first order low pass filters is given by the following relation:

$$f_c = \frac{1}{2 \cdot \pi \cdot R} \quad (4.6)$$

where R is the resistor value and C the capacitor value of the filter. For these filters the cut-off frequency has been fixed to 3120 Hz. The capacitors are ceramic capacitors which have better response at higher frequencies than electrolytic or tantalum capacitors. Having two filter stages will increment the slope of the overall filtering to -40 dB/dec from the cut-off frequency.

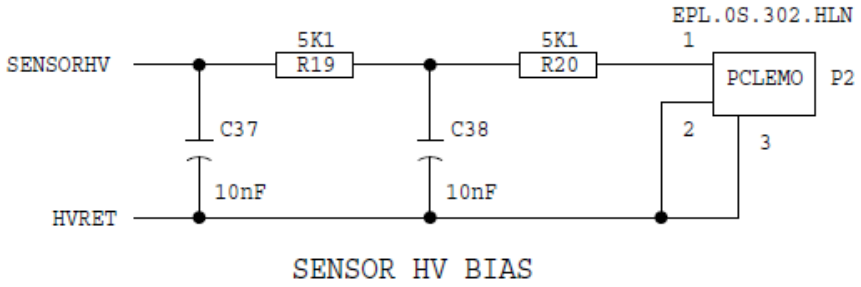

Figure 4.14. *Schematic diagram of the daughterboard detector bias system.*

4.6. Pitch adaptors

The purpose of the pitch adaptors in the daughter board is to provide an electrical connection between the analogue inputs of the Beetle chips and the microstrips of the detectors. The pitch adaptors are basically aluminum on glass structures, which contain pads and tiny tracks for connecting electrically the chip's pads and the detector's pads.

Usually, the connections between chip's pads (or detector's pads) and pitch adaptors pads are made using tiny wires which must be placed and connected (wire-bonded) using specialized wire-bonding machines. The chips, the detector(s) and the pitch adaptors are attached to the boards by means of glue. This glue can be

conductive if an electrical connection between the bottom of the chip or the detector and the board is required. For the pitch adaptors the glue is non-conductive to avoid short circuits.

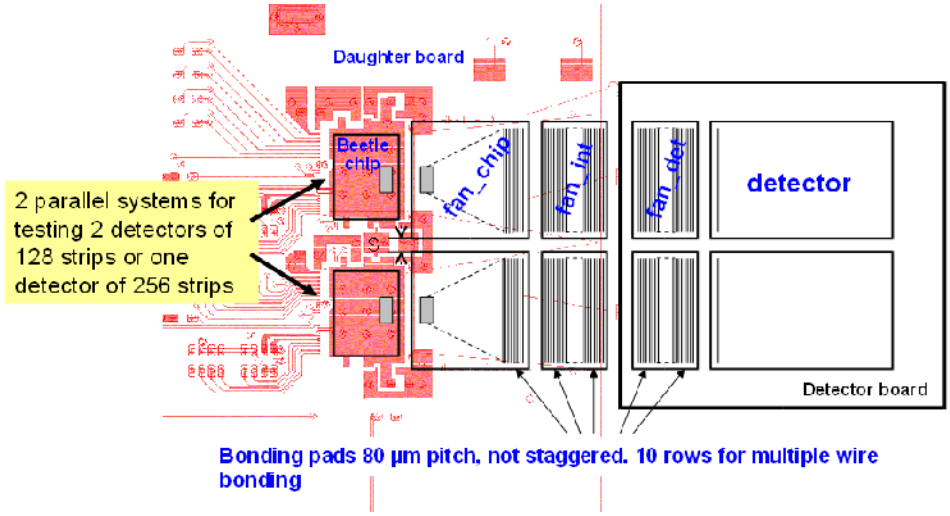

Figure 4.15. *Schematic diagram of the pitch adaptor system for daughter board.*

The design of the pitch adaptors for the daughter board, which has been carried out at the *Centro Nacional de Microelectrónica* of Barcelona, is shown in the figure 4.15. There are three pitch adaptors, one for the chip (*Fan_chip*), an intermediate pitch adaptor (*Fan_int*) and a pitch adaptor for the detector (*Fan_det*). Each pitch adaptor has pads of 80 um pitch not staggered and 10 rows of pads for multiple wire bonding. *Fan_int* and *Fan_det* are the same with this design. There could be another design of *Fan_det* for 50 um pitch detectors.

The purpose is to use the system with different kind of detectors so multiple wire bonding is necessary. For achieving this purpose, the detector(s) will be accommodated in another board different from the daughter board. Both boards are fixed to a test box by means of screws in order to have mechanical support for the bonds. The design of these boards and the test box is explained in section 4.7. With this design, the detector(s) can be changed by replacing the detector board and by re-bonding while maintaining the same daughter board. This is the reason because multiple rows of pads have been introduced in the pitch adaptors. In figure 4.16 a detailed view of the chip pitch adaptor (*Fan_chip)* can be seen. The initial 40 µm pitch of the Beetle chip input pads are broadened to the 80 µm pitch.

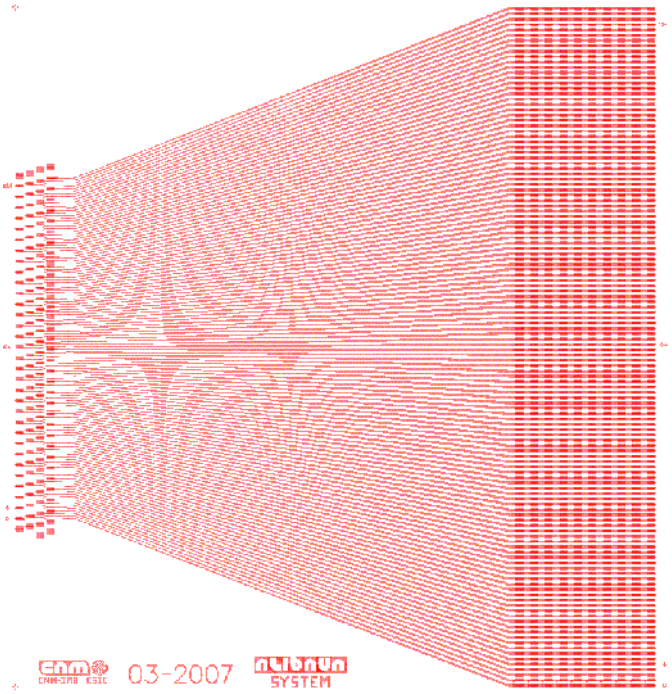

Figure 4.16. *Detailed schematic diagram of the pitch adaptor for the chip (fan-chip).*

4.7. PCBs layout and the *test box*

Two PCBs (printed circuit boards) are used to accommodate the detector(s) and the front-end readout electronics of the system. One is the daughter board, which integrates the Beetle chips and the hardware required. The other is the detector board, which can accommodate one or two detectors. The daughter board PCB is a four layer board of 72 mm x 76.2 mm designed at the *University of Liverpool*. A picture of a daughter board and a detector board attached to the *test box* is shown in figure 4.17. The top and the bottom layers of the daughter board are signal layers. The second layer from the top is the ground layer and the third layer is the power layer. There are components only on the top layer.

The ground layer has a split grounds plane for analogue and digital grounds. These grounds are connected close to the IDC connector. The detector bias ground has its own ground plane. The detector bias ground and the analogue ground are connected at the pitch adaptors. The power layer has three different power planes for the 2.5 V digital supply, 2.5 V analogue supply and the 3 V analogue supply. All the power supply levels are decoupled by means of bypass capacitors close to the

power supply pins of the chips in order to minimize the supply noise produced by the digital circuits of the Beetle chips and to isolate the analogue circuits from the supply noise.

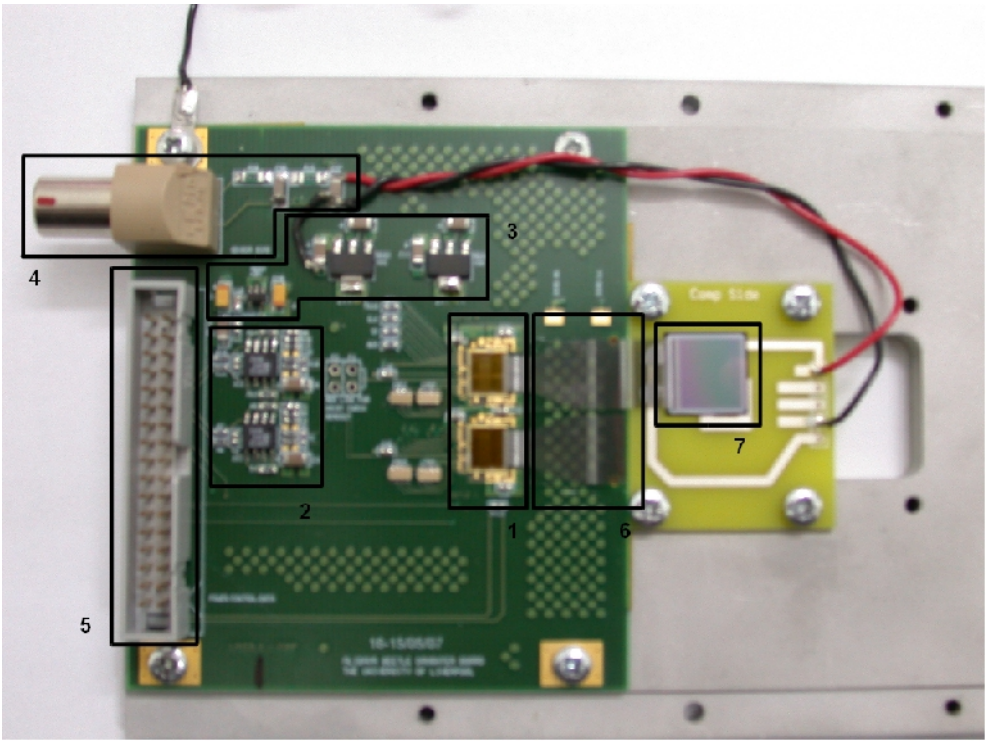

Figure 4.17. *Picture of the daughter board and the detector board (with a detector glued) screwed to the base plate of the test box. Blocks marked on the daughter board: Beetle chips (1), analogue buffers (2), hardware power supply system (3) and detector bias system (4), male IDC connector (5) and pitch adaptors (6). The detector (7) is marked on the detector board.*

The signal distribution at the IDC connector has been carefully devised for minimizing the crosstalk and the noise radiation as well as for increasing the noise immunity at the flat ribbon cable. Two different techniques have been followed to achieve this. The signals have been grouped according their nature, *i.e.* fast LVDS digital signals, slow I^2C digital signals, analogue signals, and power signals for minimizing the crosstalk between signals. Moreover analogue ground and digital ground signals have been distributed in order to isolate signals of different nature and to minimize the loops as it is shown in figure 4.18. On the other hand, different signal traces have controlled impedance for maintaining signal integrity by matching impedance. These signals have been treated as microstrip lines because

of the PCB layer structure. In particular, the LVDS differential signals (100 Ω) and differential analogue output signals (100 Ω) have traces with controlled impedance.

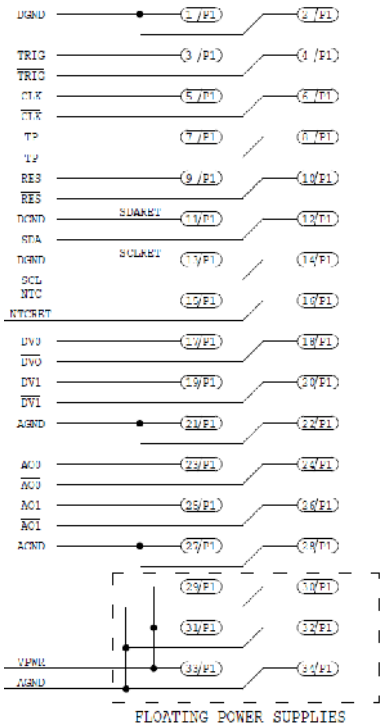

Figure 4.18. *Schematic diagram of the signal distribution at the IDC connector.*

There are two different flavours for the detector(s) boards. One of these PCBs has been designed for smaller microstrip sensors and the other one for larger microstrip sensors. In figure 4.19 these two types of detector boards are shown. The right side PCB is a board of 37 mm x 50 mm and it is for larger microstrip detector(s) with maximum dimensions of 1 cm x 3 cm. The left side PCB is a board of 37 mm x 32 mm and it is for smaller PCB detector(s) with maximum dimensions of 1 cm x 1 cm. Both PCBs have connections for the detector bias voltage and for gluing the detector pitch adaptor.

A *test box* has been designed for accommodating the daughter board and the detector board. This *test box* acts as a mechanical support for both boards facilitating the wire-bonding. The boards are fixed to a base plate using screws. A conceptual drawing of the *test box*, the daughter board and the detector(s) board is shown in figure 4.20. Both the detector(s) boards and the *test box* have been also designed at the *University of Liverpool*.

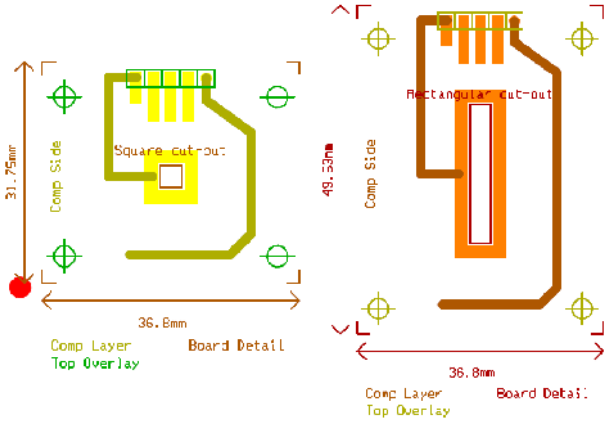

Figure 4.19. *Detector boards detail diagram. Left side diagram corresponds to the board for smaller (1 cm x 1cm) microstrip detector(s). Right side diagram corresponds to the board for larger (1 cm x 3 cm) microstrip detector(s).*

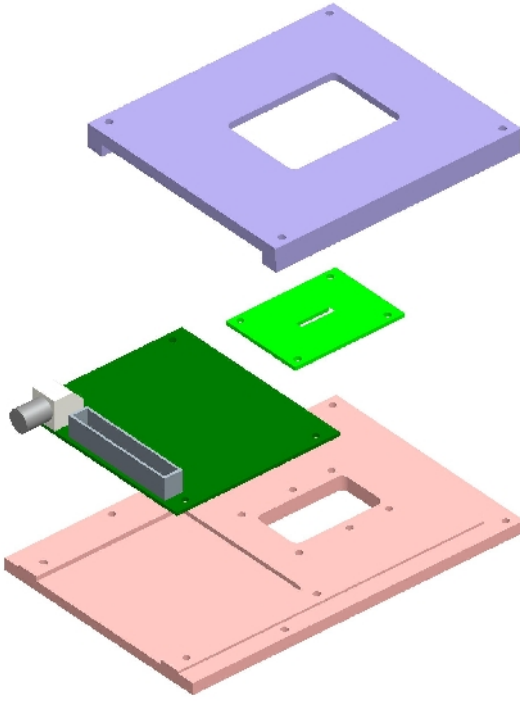

Figure 4.20. *Conceptual drawing of the test box, the daughter board and the detector(s) board. Courtesy of Ashley Greenall.*

Chapter 5

The mother board

The design of the mother board is treated in this chapter. In section 5.1, the block diagram of this mother board is explained in order to have an overall view of the board hardware. The different hardware blocks of the block diagram are described with more detail in section 5.2. In section 5.3, the FPGA logic block diagram is presented. Then, the detailed design of each logic block implemented in the FPGA is explained. In section 5.4, the firmware designed for the embedded processor implemented in the FPGA is described. This firmware has been structured as a Finite State Machine (FSM) and the different states of that FSM are described in depth.

5.1. Mother board block diagram

As it has been explained in chapter 3, the mother board is intended to process the analogue data sent from the Beetle readout chips as well as to process the trigger input signal (radioactive source setup) or to generate a trigger signal (laser setup). It is also used to control the whole system and to communicate with a host computer via USB.

The block diagram of the motherboard is shown in figure 5.1. The main component of the system is a FPGA which implements the digital logic for controlling the rest of the blocks. The S*ignal Conditioning* block is intended for transforming the differential voltage analogue input signal (one from each Beetle chip) in order to drive an oscilloscope, which requires a single ended signal, and an *analogue to digital converter* (ADC), which requires a differential input shifted

signal. The thermistor signal coming from the daughter board is digitized at the digital converter block.

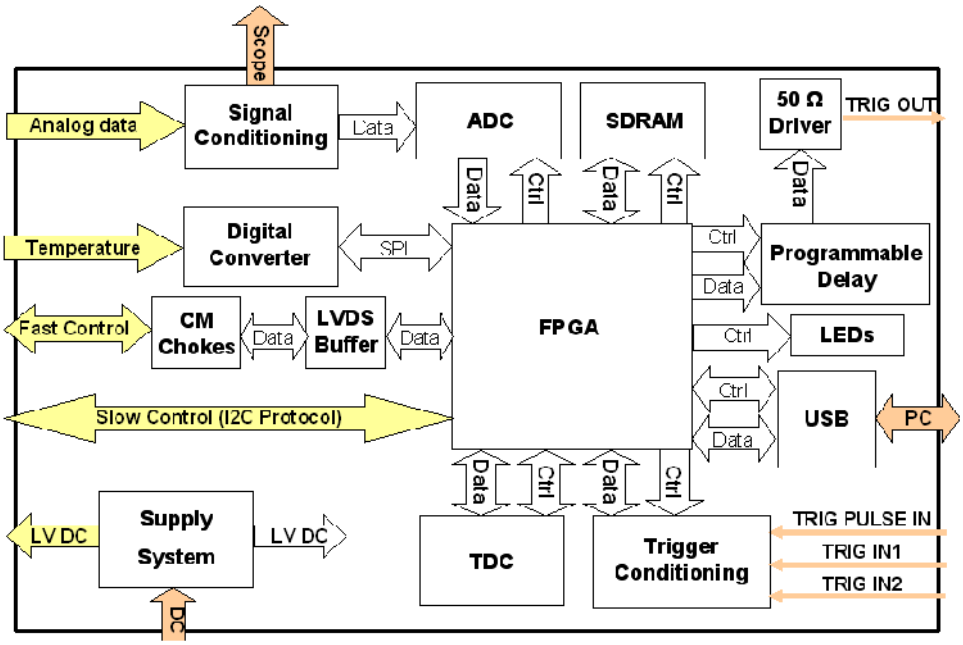

Figure 5.1. *Block diagram of the mother board.*

The *fast control* signals are generated by the FPGA. The FPGA can manage directly the *fast control* LVDS signals of the Beetle chip but, in order to protect this component, these signals have been buffered with two LVDS repeaters, one for the output signals to the daughter board (*Clk*, *Testpulse*, *Trigger* and *Reset*) and another one for the input signals from the daughter board (*DataValid0* and *DataValid1*). A common mode noise suppressor choke have been also provided for each signal. The I²C *slow control* signals are generated directly by the FPGA. There is also a SDRAM (*synchronous dynamic random access memory*) of 256 Mbits. The function of this memory is to temporally store the digitized data in each acquisition prior to be read by the software.

When the radioactive source setup is used, a trigger from an external source should be processed by the system. This input trigger can come from one or two photomultipliers (TRIG IN1 or TRIG IN2), alternatively it can be a positive/negative voltage pulse up to ±5V or fast negative NIM (TRIG PULSE IN). From this inputs, the *Trigger Conditioning* block will generate four signals in LVPECL (*low voltage positive emitter coupled logic*) format indicating if TRIG IN1 or TRIG IN2 is active (with a leading edge discriminator), or if a negative or a

positive pulse have been received on TRIG PULSE IN (with two discriminators).

The four discriminators are implemented with two dual LVPECL high speed comparators. Four voltage thresholds are needed in this block, two of them are the discrimination levels for TRIG IN1 and TRIG IN2, and the other two are the discrimination levels for positive and negative pulses (TRIG PULSE IN). These voltage thresholds are programmable by the user trough a DAC (*digital to analogue converter*) of 12 bits. The *TDC* (*time to digital converter*) block is used with the radioactive source setup. This block is composed of a TDC integrated circuit with a resolution of 600 ps and a dynamic range of 100 ns. This TDC measures the time elapsed between a start signal, generated if TRIG IN1 and/or TRIG IN2 are active, or TRIG PULSE IN is active, and a stop signal generated from a periodic signal with a period of100 ns (time window for the analogue pulse shape).

In case of the laser setup, a synchronised output trigger signal (TRIG OUT) is generated in the FPGA to drive a laser source so that the pulse shape can be reconstructed. For this reason, a programmable digital delay circuit is used. With this circuit TRIG OUT can be delayed up to 255 ns in 1 ns steps by a 8 bits parallel programming code from the FPGA. Following this block a 50 Ω driver has been designed for driving a pulse generator input that triggers the laser source.

For communicating with the host computer the *USB* block is used. This block has a USB controller for USB to FIFO (*first in first out*) parallel (8 bits) bidirectional data transfer. Two LEDs (*light emitting diode*) are provided to inform about the system state to the user.

The supply system has been centralized on the mother board. A *DC* input level (5 V) is generated with a portable *AC-DC* adapter. From this *DC* level the digital levels (1.2 V, 2.5 V and 3.3 V) for the mother board are generated by means of two *DC-DC* converters and a LDO regulator. The power supply analogue levels (5 V, -5 V and 3.3 V) for the mother board are also generated with another *DC-DC* converter and a LDO regulator. The daughter board supply level (5 V) is generated by another *DC-DC* converter.

5.2. Mother board hardware

Once the block diagram of the motherboard has been described, the mother board hardware blocks and the PCB design will be discussed with more detail in the following sections.

5.2.1. Signal conditioning and ADC

There are two signal conditioning blocks and two ADCs in parallel, each one for the corresponding analogue input signal from each Beetle chip. These analogue signals are buffered at the daughter board and converted to voltage differential signals. The signal conditioning block and the ADC block diagram for one analogue input signal can be seen in figure 5.2.

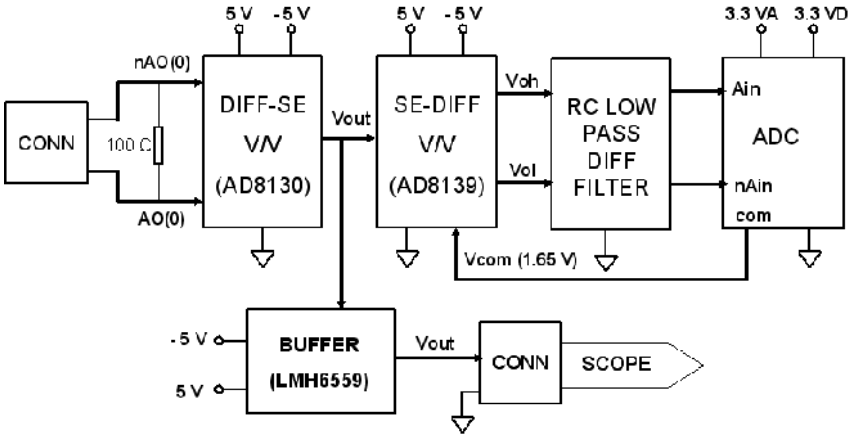

Figure 5.2. *Block diagram of the signal conditioning and the ADC designed for each Beetle analogue differential signal.*

The analogue input differential signal from each Beetle chip is terminated with a 100 Ω resistor for matching the line impedance in order to optimize the signal integrity (see chapter 4). The analogue differential voltage signal at the resistor will have a common mode voltage of 1.5 V since this signal has been shifted at the analogue buffer of the daughter board. The differential voltage range will depend on the data of the signal although it could be considered lower than ± 512 mV over the common mode voltage (beyond these limits the analogue buffers of the daughter board will saturate).

The first stage in the signal conditioning of the two analogue input signals is a differential to single-ended voltage amplifier with unity gain. The AD8130 [55] of *Analog Devices* has been used for this purpose. The AD8130 is stable at unity gain and can be used for applications where lower gains are required. The CMMR (Common-Mode Rejection Ratio) of a differential amplifier is given by

$$CMRR = 20 \cdot \log\left(\frac{A_d}{A_c}\right) \quad (5.1)$$

where A_d is the differential gain and A_c is the common-mode gain. The AD8130 has a CMMR higher than 50 dB at 40 MHz. The bandwidth with unity gain is 240 MHz. The input differential impedance is very high both for differential mode and common mode (6 MΩ and 4 MΩ respectively). The slew-rate of this amplifier is 950 mV/ns which is high enough for the input signal (which has an estimated rise and fall times of 4 ns as explained in section 4.3). The output noise of this amplifier with the configuration used can be estimated as 0.2 mV while the output offset voltage is lower than 3.5 mV. The differential input of this amplifier corresponds to the differential voltage at the 100 Ω matching resistor while the input common-mode of the amplifier is grounded in order to have a bipolar signal at the amplifier output.

The output single-ended signal is connected to two different blocks: a buffer for driving an oscilloscope input and a single-ended to differential voltage amplifier. The buffer used in this system is the LMH6559 of National Semiconductor [56]. This is a closed-loop unity gain buffer with a bandwidth of 1050 MHz and an input impedance of 200 kΩ. The output resistance is 1.2 Ω and the buffer has an output current capability up to ± 74 mA. This is optimum for driving an oscilloscope input of 50 Ω using a coaxial cable with a bipolar signal whose dynamic range is ± 512 mV. The slew-rate of this buffer is 4580 mV/ns, sufficient considering the characteristics of the signal that will be buffered. The output noise of this buffer will be about 0.2 mV. The buffer output is connected to a standard *Lemo* connector.

The single-ended to differential amplifier used for driving the differential ADC inputs is the AD8139 [57] of *Analog Devices*. The input signal of this amplifier is the single-ended signal from the first amplifier. The amplifier has unity voltage gain, which is fixed by means of two 200 Ω resistors. Therefore, the input impedance of the amplifier stage is 267 Ω. The unity gain bandwidth is 240 MHz and the slew-rate is 800 mV/ns. With this configuration the differential output noise is about 0.1 mV and the output offset voltage is lower than 7 mV. The CMRR with this amplifier is 45 dB at 40 MHz.

The single-ended bipolar input signal will also be shifted choosing among three different voltage references from the ADC depending on the full-scale range desired. This is carried out by means of a three position rotary switch whose three inputs are connected to the three different ADC reference levels and its output is connected to the common mode input of the differential amplifier.

The aforementioned amplifiers and the buffer are biased with 5 V and -5V since their performance is better for dual supply than for single supply and dedicated

analogue 5 V and -5 V are required for the discriminators in the *Trigger Conditioning* block. The power supply inputs of each one of the three blocks are bypassed with a combination of two parallel capacitors (100 nF and 10 μF) and a series ferrite bead (4S4 material with a *DC* resistance of 450 mΩ and a impedance of 600 Ω at 100 MHz) in order to maximize the noise immunity at high and low frequencies.

The ferrite is a component whose impedance varies with the frequency, having very low impedance at low frequencies and increasing its impedance at higher frequencies. Therefore is a good solution for suppressing noise from low impedance noise sources like supply systems [47-49]. The insertion of a ferrite bead in series, in addition with the decoupling capacitors, will change the impedance matching of the noise source (increasing this source impedance) and the noise victim (decreasing the victim impedance), protecting the victim from high frequency noise. Therefore, the high frequency noise is converted into heat rather than reflected to the source. Because of the bead ferromagnetic material, the ferrite bead will saturate at a current limit, so it has to be chosen for an operation under this limit (200 mA for the ferrite beads used in this case).

Prior to the ADC, a differential RC low pass filter (figure 5.3) has been placed in order to suppress some of the wide-band noise associated with high speed amplifiers. The resistors and capacitors values of the filter will depend on the cut-off frequency (-3 dB) of the filter and the single-ended to differential amplifier output characteristics, as capacitive load capability and ringing-oscillation of the output. For this system a resistor value of 51 Ω and a capacitor value of 22 pF have been chosen with a resulting cut-off frequency of 142 MHz. Therefore, it is prevented the ringing-oscillation of the outputs with the resistor and a first order low pass filter is implemented with the capacitor.

The ADC used in this system is the MAX1448 [58] of Maxim-Dallas. It is a 10-bit flash type ADC that is operated with a sample rate of 40 MHz. The input dynamic range is set by means of an internal reference at the ADC (V_{REF} = 2048 mV at *REFOUT* pin). This V_{REF} can be reduced by means of a resistive divider to the required full-scale range or it can be applied directly by means of a 10 kΩ resistor to the REFIN pin. Depending on the reference voltage at the REFIN pin, the ADC generates three output voltage references V_{REFP}, V_{REFN} and V_{COM} given by

$$V_{COM} = \frac{V_{DD}}{2} = 1650\,mV, \quad (5.2)$$

$$V_{REFP} = \frac{V_{DD}}{2} + \frac{V_{REFIN}}{4} = 2162\,mV, \quad (5.3)$$

$$V_{REFN} = \frac{V_{DD}}{2} - \frac{V_{REFIN}}{4} = 1138\,mV \quad (5.4)$$

where V_{DD} is the analogue power supply level of the ADC (3.3V). These voltage references are used as inputs for the rotary switch. Therefore, the input analogue signal can be shifted with these three values. The initial full-scale range of the ADC is given by V_{REFIN}, in this case V_{REFIN} is 2048 mV so the full-scale range is ±1024 mV over 1650 mV. If V_{REFP} is chosen as reference voltage for shifting the analogue signal, the full-scale range would be doubled just for negative signals. Conversely, if V_{REFN} is used as the reference, the full-scale range is doubled only for positive signals. The ADC has a 10-bit signed code, *i.e.* 9-bit code plus one sign bit. Therefore, the nominal resolution of the ADC will be

$$1\,LSB = \frac{V_{REFP} - V_{REFN}}{2^9} = 2\,mV \quad (5.5)$$

where 1 LSB is the value of the less significant bit or the resolution. Regarding the static parameters of the ADC [59], the integral non-linearity (INL) is lower than 2.2 LSB following the straight line method and the differential non-linearity (DNL) is lower than 1 LSB, so there are no missing codes. Regarding the dynamic parameters of the ADC [59], the maximum signal to noise ratio (SNR) for a waveform perfectly reconstructed from the digital samples is given by

$$SNR_{MAX} = (6.02 \cdot n + 1.72)\,dB \quad (5.6)$$

where n is the ADC resolution (9 bits for this ADC). This is the maximum *SNR* considering just the ADC quantization noise. For this ADC, this maximum SNR will be 56 dB (the ADC quantization noise will be 631 times lower than the signal) which is sufficient. The aperture delay (between the falling edge of the sampling clock and the instant when the actual sample is taken) is 1 ns and the aperture jitter (sample to sample aperture delay variation) is 50 ps (rms). Finally, the ADC output noise will be 0.2 LSB, which corresponds to 0.4 mV for a 2 mV/LSB.

The ADC has two separate power supplies for the analogue pins and the digital interface. The analogue supply V_{DD} is 3.3 V. The analogue supply inputs are bypassed with two parallel capacitors (100 nF and 10 μF) and a series ferrite (4S4 material with a *DC* resistance of 450 mΩ and a impedance of 600 Ω at 100 MHz).

The digital power supply is also 3.3V and is decoupled with two parallel capacitors (100 nF and 2.2 μF) in order to avoid the digital power spikes to circulate throughout the power plane [47-49]. The 100 nF capacitor is ceramic whose frequency response is better at higher frequencies and the 2.2 μF capacitor is a tantalum capacitor with low ESR (*equivalent series resistance*) for lower frequencies.

The digital interface of the ADC is a 3.3 LVCMOS (Low Voltage CMOS), which is compatible with the FPGA input-output ports. The digital signals used in this design are the following

- PD (*power down*) input. When high the ADC will be in power down mode consuming about 35 μA. When low the ADC is ready for normal operation.
- D0-D9 outputs: digital data outputs.
- CLK: sampling clock input.

The sampling clock has a frequency of 40 MHz. The samples are taken with the falling edge of the sampling clock while the output data is valid on the leading edge of the sampling clock. There is an internal latency of 5.5 clock cycles from the data is sampled until the data is valid at the output (figure 5.3).

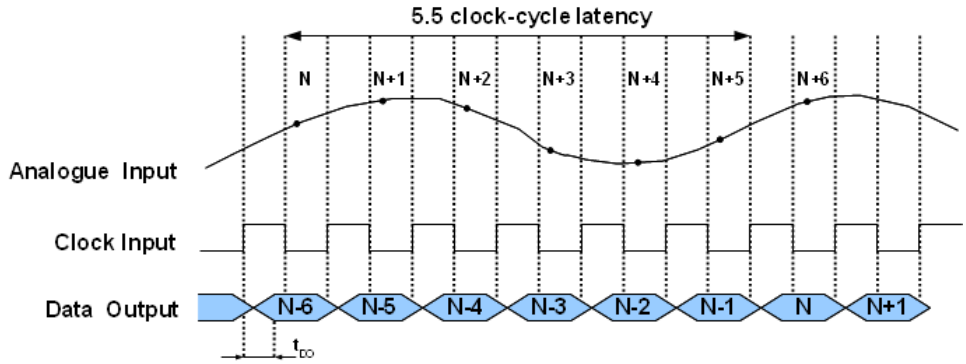

Figure 5.3. *ADC MAX1448 timing diagram.*

There is no need of an anti-aliasing low pass filter in order to comply the *Nyquist-Shannon* theorem [60] since in this system the analogue input signal is not sampled continuously to reconstruct it. Only a specific point in each one of the 128 multiplexed channels of the two analogue input signals coming from the Beetle chips is interesting for this system (see chapter 4 for a detailed explanation of the Beetle analogue output signal format). Therefore, the analogue input signals from the Beetle chips will not be reconstructed from the acquired samples but the system

will store the voltage value of these samples (proportional to the charge collected in each microstrip of the corresponding silicon detector).

5.2.2. *Fast control* and *slow control*

The fast control signals for operating the Beetle chips (*Clk*, *Testpulse*, *Reset* and *Trigger*) are generated by the FPGA. *DataValid0* and *DataValid1* input signals coming from the daughter board are connected to FPGA as well. These input signals are active when a Beetle analogue readout is taking place. All these signals follow LVDS [43, 52] standard, with the advantages of this format discussed in chapter four.

The fast control output signals are generated with the FPGA in 3.3V LVCMOS format. They are connected to a LVDS line driver. This is the DS90LV047A [61] of *National Semiconductor*, a quad LVCMOS-LVTTL/LVDS converter. It accepts up to four input signals with LVCMOS or LVTTL format and it converts them into ± 350 mV LVDS signals. The maximum bandwidth of the input signals is 200 MHz, enough for this design. The maximum propagation delay of this converter is 1.7 ns with 400 ps of maximum differential skew. The converter has been biased with a digital 3.3 V supply level. This supply level is decoupled with three capacitors in parallel (1 nF, 100 nF and 33 μF) for reducing the peak current surges generated by the converter and having a stable power supply level.

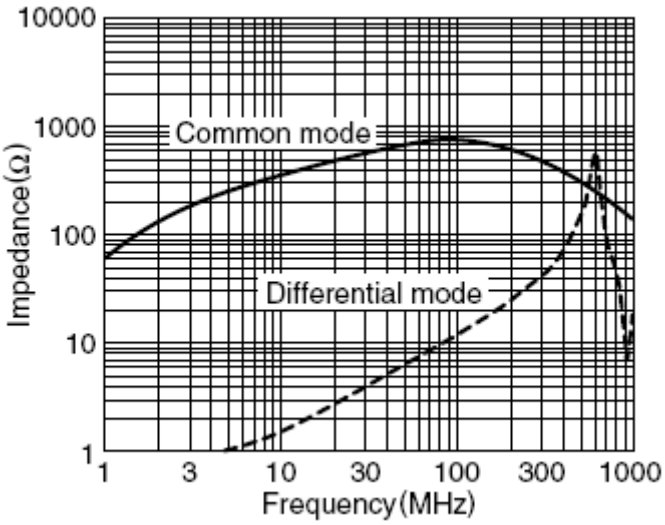

Figure 5.4. *ZJYS51R5-4PT-01common mode filter impedance response for common mode and differential signals. Figure taken from [62].*

Once the *fast control* output signals have been converted into LVDS format, they are connected to two common mode filters. These are the ZJYS51R5-4PT-01 [62] of *TDK*. These common mode filters are chokes for suppressing the common mode noise in differential lines. This type of filters works by increasing the common mode impedance while having low differential mode impedance [63]. The common mode current, flowing in the same direction through each of the choke windings, creates equal and in-phase magnetic fields which add together. The differential mode current, flowing in opposite directions through the choke windings, creates equal and opposite magnetic fields which cancel each other out. This results in the choke presenting a high impedance to the common mode signal, which passes through the choke heavily attenuated, and low impedance to the differential mode signal, which passes through the choke without attenuation. The actual attenuation (or common mode rejection) depends on the relative magnitudes of the choke impedance and the load impedance. The load impedance of the output LVDS signals will be 100 Ω between the differential lines at the daughter board (this termination is required for LVDS signals, see chapter four for more details), while the choke impedance for this filter is shown in the figure 5.5.

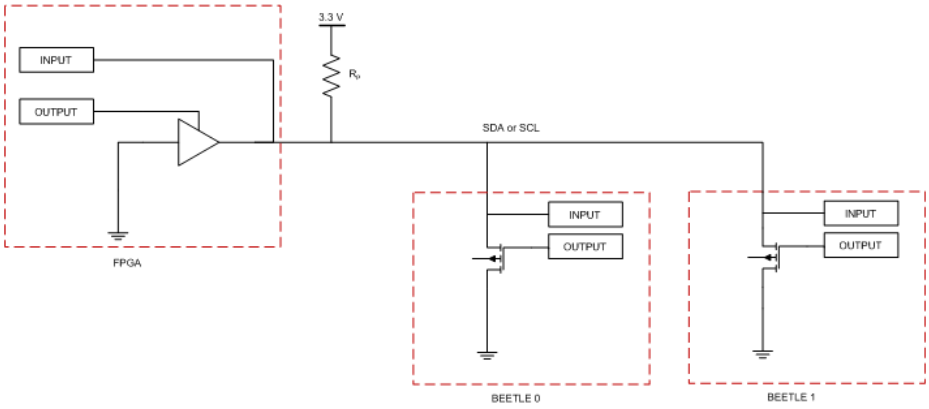

Figure 5.5. *Implementation scheme of SDA and SCL bidirectional ports by using tri-state ports at the FPGA.*

The *fast control* LVDS input signals coming from the daughter board (*DataValid0* and *DataValid1*) are connected to another common mode filter of the same type as those used for the other fast control signals. Therefore, here apply the same considerations. After the common mode filter the signals are terminated each one with a 100 Ω resistor between the differential lines as requires the LVDS standard. Then, these two LVDS signals are converted into 3.3 V LVCMOS signals using the DS90LV048A [64] quad LVDS line receiver of *National Semiconductor*.

This line receiver has a maximum bandwidth of 200 MHz, a propagation delay of 2.7 ns and a maximum differential skew of 400 ps. The converter has been biased with a digital 3.3 V supply level. This supply level is decoupled with three capacitors in parallel (1 nF, 100 nF and 33 μF) for reducing the peak current surges generated by the converter and having a stable power supply level. The two 3.3 V LVCMOS output signals are connected to the FPGA.

The *slow control* bidirectional signals, SDA (*serial data line*) and SCL (s*erial clock line*), are generated directly with the FPGA which acts as master transmitter and receiver. Therefore, these two inputs/outputs must have an open-drain or open-collector to perform the wired-AND function. These inputs/outputs have been implemented using two tri-state buffers of the FPGA, as can be seen in figure 5.5.

Moreover, both bidirectional I²C lines must be connected to a common V_{DD} level trough two pull-up resistors, R_p. This V_{DD} level has been fixed to 3.3 V, which is compatible with both the FPGA and the Beetle chips. The pull-up resistors value depends on the supply voltage, the number of connected devices and the bus capacitance. First, assuming that just one device puts the bus on low level, the minimum value for the R_p resistor is given by

$$R_{pMIN} = \frac{V_{DD} - V_{OL}}{I_{OL}} \quad (5.7)$$

where V_{OL} is the low level output voltage for a I²C port (open-drain or open-collector) and I_{OL} is the sink current at the I²C port at V_{OL}. From the I²C specification [42], V_{OL} = 0.4 V and I_{OL} = 3 mA, V_{DD} = 3.3 V, then R_{pMIN} = 966 Ω. Second, the maximum value of R_p resistor is given by

$$R_{pMAX} = \frac{0.8 \cdot V_{DD}}{I_{L1} + I_{L2} + I_{L3}} \quad (5.8)$$

where I_{L1}, I_{L2} and I_{L3} are the maximum high level input currents at each connected device port (FPGA, Beetle 0 and Beetle 1). From the I²C specification, $I_{L1} = I_{L2} = I_{L3}$ = 10 μA and V_{DD} = 3.3 V, then R_{pMAX} = 88 kΩ. Finally, the bus capacitance constrains the maximum value of the pull-up resistor. From the I²C specification, in the worst case (the maximum bus capacitance) R_{pMAX} = 2 kΩ. Therefore, the value of the pull-up resistors for the SDA and SCL lines has been fixed to 1 kΩ. These two pull-up 1 kΩ resistors have been placed close to the FPGA bidirectional ports for the SDA and SCL lines. The 3.3 V supply level is decoupled with three capacitors in parallel (100 nF, 100 nF and 33 μF) for reducing the peak current

surges generated by lines and having a stable power supply level.

5.2.3. Trigger input blocks (discrimination, DAC and TDC)

The trigger input blocks are used with a radioactive source setup for obtaining a time stamp of each trigger. There are three different trigger inputs. Two of them are signals from photomultipliers (TRIG IN1 and TRIG IN2) and the other one (TRIG PULSE IN) is an auxiliary signal that can be a fast NIM (*nuclear instrumentation module*) logic signal or voltage pulses up to ±5V. Firstly, these signals must be converted to common digital signals. Then, the user should be able choose between using as trigger the input signals of the photomultipliers or the auxiliary signal. If the photomultiplier signals are used, the user may select whether a coincidence of both photomultiplier signals will be used as trigger or just one of them will be used as a trigger. The resulting signal will be the trigger signal which will be used for driving a TDC with other periodic signal as a reference.

The fast NIM logic levels are defined in [65]. The low logic level is defined as ±2 mA (or ±100 mV over 50 Ω) while the high logic level is defined as -12 mA to -32 mA (or -600 mV to -1600 mV over 50 Ω). The rise and fall times of the fast NIM logic signals are not defined in the standard, but typically a 2 ns is considered. The input signal from a photomultiplier is an analogue negative pulse. The fall time of this pulse is constant and generally lower than 5 ns whereas the amplitude can vary from one pulse to another one. However, this amplitude can be considered lower than 100 mV.

One of the purposes of these input blocks is to convert the input trigger signals into digital signals with minimum delay and, above all, minimum delay dispersion. For the input signals from the photomultipliers, a leading edge discrimination is carried out. Given an input pulse, the leading edge discriminator produces an output pulse when the input pulse crosses a given threshold voltage. The main problem associated to leading edge discriminators is that if the amplitude of the input pulse changes while keeping constant the rise time a time-walk occurs, *i.e.* for pulses with the same rise time those with smaller amplitude cross the threshold later (homothetic signals).

Another option would be to use a *constant fraction discriminator* (CFD) [17, 18], it minimizes the problem of the time walk by using a constant fraction of the input pulse to precisely determine the timing of the output pulse relative to the input signal.

In this case a simple leading edge discriminator has been chosen because the

time-walk will be low considering the characteristics of the photomultiplier signals, rise time below 5 ns and amplitudes below 100 mV. The leading edge discrimination is performed using a high speed PECL (*positive emitter coupled logic*) comparator. The comparator used is the MAX9601 [66] of *Maxim-Dallas*. This is a dual comparator with PECL outputs, a propagation delay lower than 700 ps and propagation delay dispersion lower than 40 ps. The comparator has different power supplies for the input stage and for the output driver.

The power supplies for the analogue inputs, V_{CC} and V_{EE}, have been fixed to 5 V and to -5V respectively, since the analogue inputs and the thresholds will be negative in this case. The power supply for the output driver, V_{CCO}, is 2.5 V. This value has been chosen in order to have a 2.5 V LVPECL outputs since this standard is compatible with the FPGA inputs. Therefore, the comparator outputs can be connected directly to the FPGA without the need of a logic translator which would introduce more propagation delay and propagation delay dispersion. The LVPECL format is a high speed differential standard derived from the ECL standard by using a positive rail and ground as references [67]. In case of the 2.5 V LVPECL format, the reference levels are 2.5 V and ground. With these reference levels, the logic output high voltage is 1560 mV and the logic output low voltage is 780 mV, having an output swing of 780 mV and a common mode voltage of 1170 mV. The advantages of the 2.5 V LVPECL against the ECL are lower power consumption and the required reference voltage levels (ground and -5.2 V for ECL), however termination resistors are still required because the outputs are open emitters of bipolar transistors that must be connected to the lower reference level through a resistor.

The termination resistors values must be calculated properly in order to assure the output voltage logic levels and the dynamic performance for the 2.5 V LVPECL standard. The output continuous safe current limit, I_{MAX}, determines R_E minimum *DC* termination scheme resistance to ground although this will not provide a practical *AC* signal termination [68]:

$$R_{EMIN} = \frac{V_{OHMAX}}{I_{MAX}} \quad (5.9)$$

where V_{OHMAX} is the logic high maximum output voltage. In this case V_{OHMAX} = 1750 mV and I_{MAX} = 50 mA, so R_{EMIN} = 35 Ω. On the other hand, the dynamic function of the termination resistor, R_E is to develop the voltage change, ΔV, during a high to low or low to high transition and present this to the transmission medium such as coaxial, twisted pair, microstrip or stripline (figure 5.6). The ΔV signal propagates to the receiver and is either reflected, dissipated, or a combination.

When R_E is too large, steps appear in the trailing edge of the propagating signal, ΔV, at the input to the receiving gate, slowing the edge speed and increasing the net propagation delay. A reasonable negative-going signal swing at the input of the receiving gate results when the value of R_E is selected to produce an initial step of 75% of the expected ΔV, or a 600 mV step for a 800 mV signal at the driving gate [68]

$$\frac{V_{OHMIN} - V_{EE}}{R_{EMAX} + Z_0} \cdot Z_0 = 0.6 \quad (5.10)$$

where V_{OHMIN} is the is the logic high minimum output voltage, V_{EE} is the lower reference voltage and Z_0 is the impedance of the transmission line. In this case V_{OHMIN} = 1400 mV, V_{EE} = 0 V and Z_0 = 50 Ω (and therefore R_t = 100 Ω for matching the transmission line impedance), then R_{EMAX} = 66.6 Ω. Hence, a termination resistor R_E value of 51 Ω has been chosen in order to fulfill the requirements imposed.

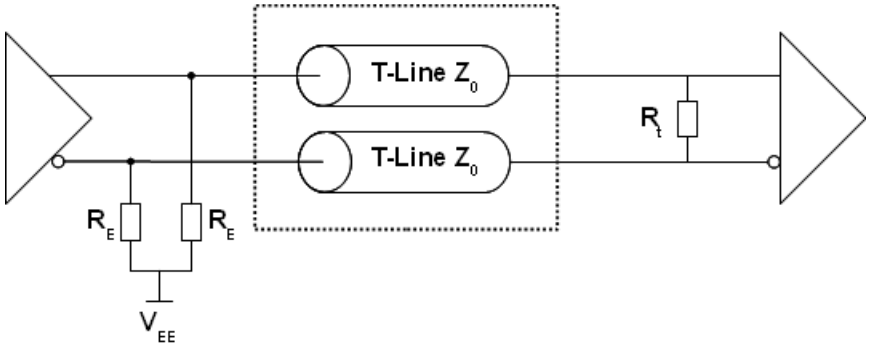

Figure 5.6. *Output termination scheme considered for the MAX9601 2.5 V LVPECL comparator output signals.*

Regarding the photomultiplier inputs (TRIG IN1 and TRIG IN2) and thresholds (THRA and THRB), the comparator must generate a positive pulse (SIN1 for TRIG IN1 and SIN2 for TRIG IN2) if TRIG IN1 (TRIG IN2) < THRA (THRB). Both the photomultiplier signals and the thresholds will be negative, the photomultipler inputs are connected to the negative input of each comparator while the threshold are connected to the positive input of each comparators. The thresholds THRA and THRB are the negative discriminator thresholds for a high logic output and they are generated by a DAC for user control and accuracy. The MAX9601 has hysteresis capability by means of connecting a resistor to the hysteresis input. For the discrimination of the photomultipliers signals there is no hysteresis in order to

minimize the time-walk. Finally, the input impedance for the photomultipliers input signals have been fixed to 50 Ω for matching the line impedance of these signals since they are transmitted using 50 Ω coaxial cable connected to standard *Lemo* connectors.

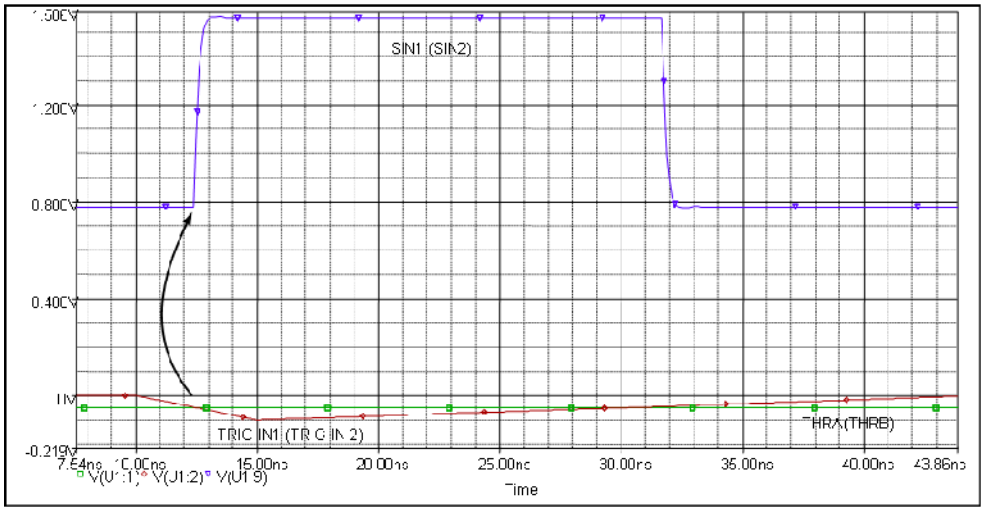

Figure 5.7. *PSpice simulation result for the leading edge discrimination of the photomultiplier input signals.*

The auxiliary input signal (TRIG PULSE IN) can be a fast NIM logic signal or voltage pulses up to ±5V. It will be also transmitted using 50 Ω coaxial cable connected to standard *Lemo* connector, so a 50 Ω input impedance is required and it has been implemented. This signal will be converted into a 2.5 V LVPECL output signal using the same dual MAX9601 comparator. In this case a hysteresis of 60 mV has been introduced by connecting a 10 kΩ resistor to the hysteresis comparator input for minimizing the possible noise in the input signal. The signal conversion has been implemented using two comparators. The auxiliary input is connected to the positive input of one comparator (with a positive threshold level, THRD, connected to the negative input) and to the negative input of the other comparator (with a negative threshold level, THRC, connected to the positive input). Therefore, there will be two 2.5 V LVPECL outputs (PPOS or PNEG) of the comparators that will warn if a positive pulse or a negative pulse has been detected depending on the threshold levels according to the following rules,

- if TRIG PULSE IN > THRD, THRC (positive pulse) then PNEG = 0 and PPOS = 1.
- If THRC < TRIG PULSE IN < THRD (logic 0): PNEG = 0 and PPOS = 0.

- If TRIG PULSE IN < THRC, THRD (negative pulse): PNEG = 1 and PPOS = 0.

A simulation result obtained with *PSpice* [69] for the photomultiplier input signal discrimination is shown in figure 5.7, whereas a simulation result obtained for the auxiliary signal level conversion can be seen in figure 5.8. In the first case, a hysteresis of 3 mV and a propagation delay of 200 ps were obtained, while in the second case an hysteresis of 65 mV and a propagation delay of 100 ps were calculated.

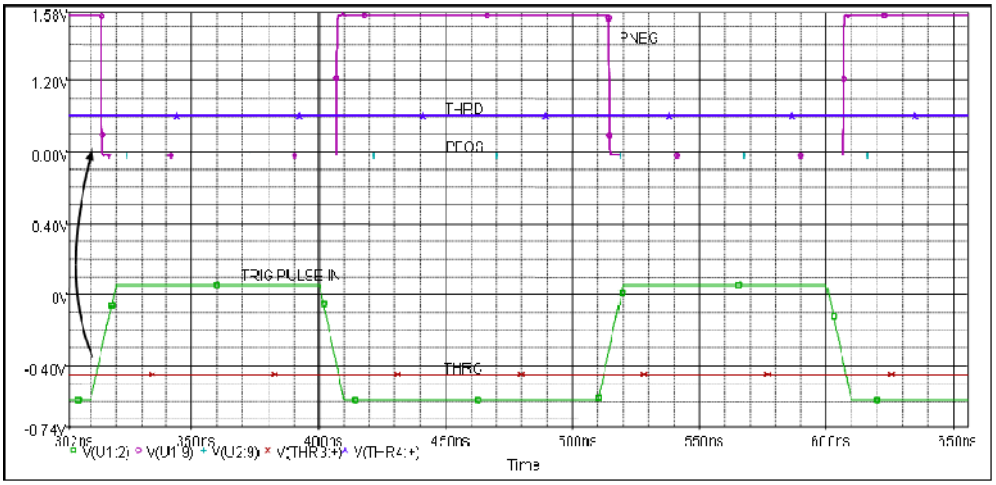

Figure 5.8. *PSpice simulation result for the auxiliary input signal (fast NIM signal) level conversion.*

The comparators have been biased using 5 V and – 5V analogue supply levels for the input stage and a 2.5 V supply level for the digital output stages. The analogue power supply inputs are bypassed with a combination of two parallel capacitors (10 nF and 10 µF) and a series ferrite bead (4S4 material with a DC resistance of 450 mΩ and a impedance of 600 Ω at 100 MHz) in order to maximize the noise immunity at high and low frequencies. The digital power supplies are decoupled with two parallel capacitors (10 nF and 10 µF) in order to avoid the digital power spikes to circulate throughout the power plane. In both cases, the 10 nF capacitor is ceramic whose frequency response is better at higher frequencies and the 10 µF capacitor is a tantalum capacitor with low ESR for lower frequencies. In figure 5.9 and figure 5.10 the schematic circuit of the *Trigger Conditioning* block is shown.

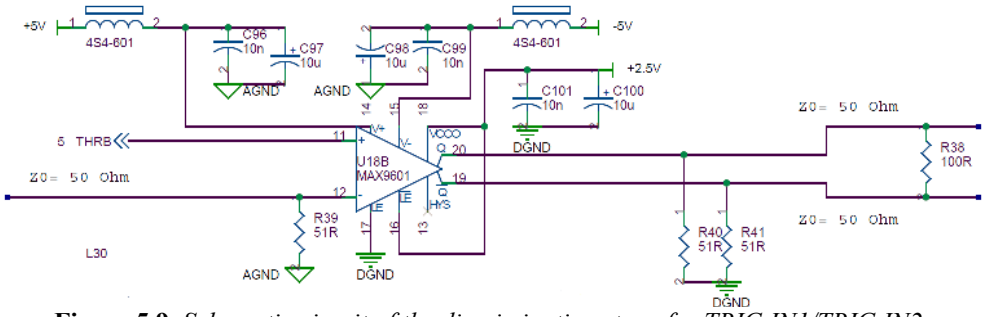

Figure 5.9. *Schematic circuit of the discrimination stage for TRIG IN1/TRIG IN2.*

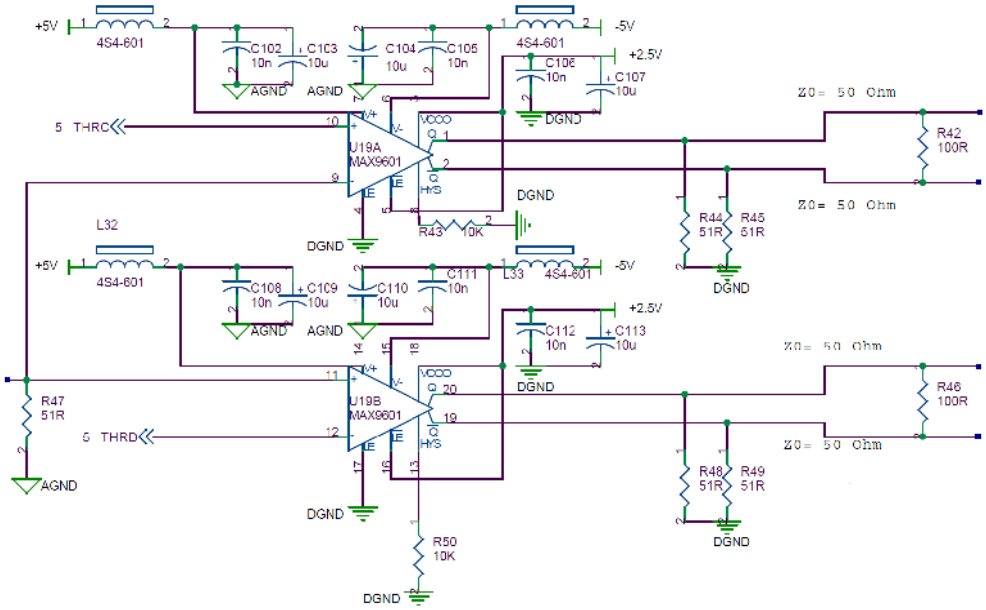

Figure 5.10. *Schematic circuit of the discrimination stage for TRIG PULSE IN.*

The discrimination threshold levels are generated using a DAC, since the user must be able to adjust the threshold levels by software and for better accuracy. The DAC used for this design is the DAC7614 [70] of Burr-Brown. This is a quad DAC for generating the four required thresholds for the photomultiplier signals discrimination (THRA and THRB) and for the auxiliary signal level conversion (THRC and THRD). The DAC7614 has bipolar output capability as negative thresholds are required. Hence, bipolar references must be used. A positive voltage reference (V_{REFH} = 2048 mV) and a negative voltage reference (V_{REFL} = -2048 mV) are implemented for this reason. Each one of the four DAC has a resolution of 12 bits. The output voltage is given by the following relation

$$V_{OUT} = V_{REFL} + \frac{(V_{REFH} - V_{REFL}) \cdot N}{4096} \quad (5.11)$$

where N is the digital input code in decimal. Therefore, considering the voltage references used and the resolution, the value of 1 LSB will be 1 mV, sufficient accuracy for this application. The output range will be limited by the voltage references, having a range of ±2048 mV at the output of each DAC. The settling time, time until the DAC output level error is ± 0.012 %, is typically 5 μs.

The quad DAC is biased with 5 V and -5 V analogue supply levels. These levels are shared both by the analogue and digital interface of the DAC. However, the block has been treated as a fully analogue block. Therefore, the power supply inputs are bypassed with a combination of two parallel capacitors (100 nF and 10 μF) and a series ferrite bead (4S4 material with a *DC* resistance of 450 mΩ and a impedance of 600 Ω at 100 MHz).

The digital interface of the DAC is a SPI (*serial peripheral interface*) bus. The SPI bus is a synchronous serial data link standard that operates in full duplex mode. Devices communicate in master/slave mode where the master device initiates the data frame. Multiple slave devices are allowed with individual slave select (chip select) lines. The DAC7614 is treated as slave. Thus, a master controller has been developed and implemented in the FPGA in order to program the DAC7614 outputs. The DAC7614 SPI is composed by three lines,

- SDI (*serial data input*) line: this is an input which will receive synchronously with the clock the data from the FPGA controller. Each data frame, will have 16 bits, first two bits for choosing the DAC to program, then 2 dummy bits and finally the 12 bits of data for configuring the voltage threshold. The data format will be big-endian (the most significant bit, MSB, first).
- CLK (*serial data clock*) line: this is the clock input for receiving the data synchronously. The maximum frequency of the clock can be 12,5 MHz.
- CS (chip select) line: active low input in order to select the chip for a data transfer.

As well as the SPI bus lines the digital interface has three more digital inputs,

- LOADDACS: active low input. The selected DAC register becomes transparent when this input is low. It is in the latched state when this input is high.

- RESET: active low asynchronous reset input. Sets all DAC registers to either zero-scale or mid-scale when it is low. RESETSEL input determines which code is active.
- RESETSEL: when low, a low level on RESET will cause all DAC registers to be set to zero-scale. When RESETSEL is high, a low level on RESET will set the registers to code mid-scale. This input is connected to 5 V supply level in this system for configuring the DAC registers to mid-scale (0 V since bipolar references are used) after a reset.

All these signals are 3.3 V LVCMOS/LVTTL compatible, so they can be connected directly to the FPGA without the aid of a level translator.

The coincidence and trigger selection logic is intended for generating a trigger pulse signal from the SIN1, SIN2, PPOS and PNEG digital signals coming from the *Trigger Conditioning* block. The user of the system must be able to select by software if the trigger pulse is generated from SIN1 or SIN2, from their coincidence or from the auxiliary signal. On the other hand, the digital pulse signals which come from the *Trigger Conditioning* block could have pulse widths lower than the TDC requirements (4 ns as it will be explained later). Therefore, it is necessary a coincidence and multiplexing logic as well as pulse generation logic in order to fulfill these requirements. These logic blocks were designed in first place in order to consider their implementation by discrete ECL logic or in the FPGA to minimize the delay dispersion, Δtd. Thus, their design and these considerations will be explained in the corresponding FPGA logic section (section 5.3.4). For the moment, it will be sufficient to know that this logic will generate a 3.3 V LVCMOS trigger pulse (TRIG) digital signal with a width higher than 4 ns from the trigger conditioning signals (SIN1, SIN2, PPOS and PNEG) taking into account the user selection and with the minimum delay dispersion.

The TDC must measure the time passed between the leading edge of a START signal and the leading edge of a STOP1 signal. The START signal will be generated by the FPGA from the TRIG pulse signal which will be active when a trigger input is detected. Therefore, the START signal will be virtually the TRIG signal. The STOP1 signal will be also generated by the FPGA from a periodic signal with a period of 100 ns and a pulse width of 25 ns if a START signal has been generated previously. The period of 100 ns is established in order to reconstruct the Beetle chip analogue pulse shape (see chapter four for a detailed description of this pulse shape).

The TDC implementation has been considered carefully. The measurement resolution requirement for the radioactive source setup should be 2 ns or tighter. It

must be taken into account that the total resolution in this case will be the delay dispersion, Δt_d, of the START and STOP1 signals and the TDC resolution itself over the whole operating range (temperature and supply level variations). This is a very conservative criterion because both the Δt_d and the TDC resolution will be worse over the whole operating range than in reality, where the mother board will operate under stable temperature and stable supply voltage levels. On the other hand, the measurement range is 100 ns, so the TDC must cope with this requirement. Finally, a minimum number of components should be used and the cost should be as low as possible. Therefore, a commercial TDC integrated circuit has been used as a first solution. However, a TDC implementation in the FPGA has been considered and it has been already developed, although its explanation is beyond the scope of this thesis.

The TDC used in this design is the TDC-GP1 [71] from *Acam mess-electronic*. It is a TDC based on CMOS propagation delays of simple logical gates (*i.e.* inverters) for fine quantization of time intervals. This TDC can be operated in different ways. In this system it is used just one of the two channels of the TDC with the STOP1 signal. The START signal is used as the common signal for starting the measurement on the START port of the TDC. Furthermore, the TDC is operated in the so-called measurement range 1. In this measurement range the following requirements are imposed,

- the dynamic range of the TDC is 7600 ns.
- The minimum pulse width for both the START and STOP1 pulses must be greater than 4 ns.
- The minimum time between a START leading edge and a STOP1 leading edge must be 3 ns. If this time is lower the TDC will not give a measurement. Therefore, in this case the FPGA TDC controller will have to detect this event and it will give a 0 ns measurement.

Each TDC measured time, t, will be a fixed point number with a signed 16 bits integer portion and a 16 bits fractional portion, related to a calibration clock period. This time is given by the following equation

$$t = \frac{[VAL(hit1) - VAL(start) - (2 \cdot CAL1 - CAL2)]}{(CAL2 - CAL1)} \cdot T_{CAL} \quad (5.12)$$

where *VAL(hit1)* is the first hit measured value (*i.e.* a leading edge of STOP1) by the TDC, *VAL(start)* is the START measured value by the TDC (zero since TDC starts measuring), *CAL1* is the calibration period measured value (corresponding to T_{CAL}), *CAL2* is calibration period measured value (corresponding to $2 \cdot T_{CAL}$) and

T_{CAL} is the calibration clock period.

This time could range nominally up to $\pm 1.99 \cdot T_{CAL}$, with T_{CAL} the period of the calibration clock. The calibration clock is derived from the TDC CLK_REF pin which is driven by a clock generated by the FPGA from the system clock. This clock has a period of 100 ns and a pulse width of 50 ns. A calibration can be carried out after each measurement in order to obtain calibrated data. In this calibration process, the values corresponding to T_{CAL} *(CAL1)* and to $2 \cdot T_{CAL}$ *(CAL2)* are measured and then used to calculate a calibrated measurement (figure 5.11). The TDC can be configured to carry out the calibration automatically after each measurement, *i.e.* after each trigger, and this is the way how the TDC will work in this system.

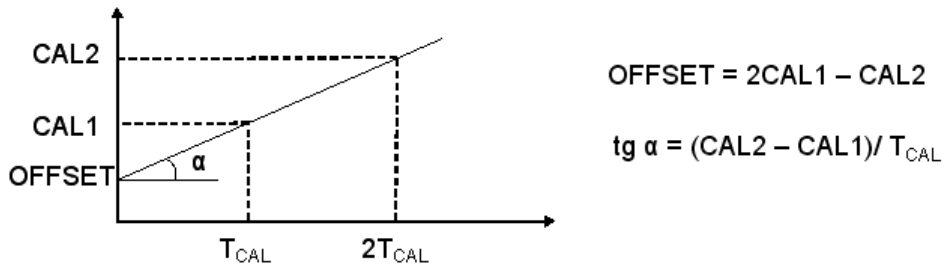

Figure 5.11. *TDC-GP1 calibration principle used for the calculation of calibrated measurements. The offset and the tgα are used in the equation 5.12. The time measured must be lower than* $2 \cdot T_{CAL}$.

The nominal resolution of the TDC (*i.e.* each bit width) is fixed by the operating conditions (supply voltage of 3.3 V at 25 °C),

- worst case resolution: 428 ps.
- Typical resolution: 306 ps.
- Best case resolution: 186 ps.

Similarly, the calculation time required from measurement until data are at TDC output will be nominally (supply voltage of 3.3 V at 25 °C),

- worst case calculation time: 8138 ns.
- Typical calculation time: 5813 ns.
- Best case calculation time: 3546 ns.

The TDC-GP1 has internal configuration registers for configuring the chip in order to be operated in a specific way (measurement range, calibration scheme,

resolution, type of measurement, etc). Moreover, the TDC has an internal ALU (*arithmetic logic unit*) for processing the measured data using the calibration data and for multiplication. There are also status registers which inform about the state of the TDC. For controlling this configuration registers and reading out the data, the TDC has a read/write digital interface with the following signals,

- ALE: address latch enable (active high). Read/write signal.
- WRN: write enable (active low). Read/write signal.
- RDN: read enable (active low). Read/write signal.
- CSN: chip select (active low). Read/write signal.
- ADR(3:0): address bus. Addresses of TDC registers.
- DATA(7:0): data bus. Bidirectional. Data to read or write.

These signals are connected directly to the FPGA ports since the TDC is controlled by the FPGA. The address bus signals, ADR(3:0), have been connected to the lower four lines of the data bus, DATA(3:0), in order to minimize the FPGA used ports. Therefore, the ALE signal will be used to select if the data sent through DATA(3:0) is an address or not.

There are other TDC signals related to the TDC measuring unit that are also connected directly to the FPGA ports,

- RST_N: reset (active low) input. All the TDC logic is initialized.
- CLK_REF: reference clock (10 MHz) input. Calibration clock (10 MHz) is derived from CLK_REF.
- START: TDC measurement unit is started when a positive edge is detected by the TDC. The START signal will be a signal derived from the TRIG signal which is generated depending on the discrimination and level conversion stages as well as the coincidence, multiplexing and pulse generation logic.
- STOP1: first measurement channel. TDC measurement unit is stopped when a positive edge is detected by the TDC (up to 4 hits per channel). The STOP1 signal will be a signal derived from an internal FPGA periodic signal with a period of 100 ns and a pulse width of 25 ns if a START signal has been generated previously.
- EN_STOP1: enable channel signal input. Active high.
- STOP2: second measurement channel. In this system, it is not used and, therefore, it is connected to low level.
- EN_STOP2: enable channel signal input. It is not used in this system and, hence, it is connected to low level.

- INTFLAG: interrupt signal output. Active high. This signal will be active when a measurement has been processed and it is ready to be read from the TDC registers.

These signals are connected directly to the FPGA ports, with the exception of STOP2 and EN_STOP2. All the described digital signals of the TDC have a 3.3V LVCMOS compatible format, so there is no need of any level conversion for connecting them to the FPGA ports.

The TDC-GP1 is biased with a digital 3.3V supply level at different TDC pins. The digital power supplies are decoupled with three parallel capacitors (100 nF, 100 nF and 33 μF). The 100 nF capacitors are ceramic and the 33 μF capacitor is a tantalum capacitor with low ESR.

5.2.4. Trigger output blocks (programmable delay line and 50 Ω driver)

The trigger output blocks are a programmable delay line and a 50 Ω driver. These hardware blocks will be used with a laser setup. The laser will be driven by a TRIG OUT pulsed signal generated by the system for pulsing the laser light through a standard pulse generator (see chapter 3 for a detailed description of the laser setup). Previously, the FPGA will generate a 3.3 V LVCMOS pulsed signal with a pulse width of 1000 ns for synchronising the system. When the system is synchronized (*i.e.* a specific position of the Beetle analogue pulse shape is sampled and acquired) this signal will be a periodic signal with a frequency of 1 KHz and the same pulse width.

In order to acquire a specific position of the Beetle analogue pulse shape, the pulses in the synchronization state must be delayed regarding the *fast control Trigger* signal which is sent to the Beetle chips for carrying out an acquisition. Once the system has been synchronized, this delay should be stored to perform a number of acquisitions with the periodic pulsed signal. This variable delay has to be specified by the user by establishing a delay range and a delay step at the synchronization state, then the system will automatically scan over the delay range. For achieving this variable delay, the FPGA will move forward the generated pulsed signal regarding the *fast control* TRIG_L signal in steps of 25 ns (as it will be explained in the corresponding FPGA logic section). Using an external programmable delay line this generated pulsed signal will be delayed (*i.e.* moved in the inverse direction) in steps of 1 ns.

The programmable digital delay line used in this system is the 3D7428-1 [72] from *Data Delay Devices*. This is a 8 bit digital programmable delay line with a resolution of 1 ns and delay range of 255 ns. This delay line has an inherent delay of 11.5 ± 2.5 ns, that is, with a programmed zero delay the output signal will have a delay of 11.5 ± 2.5 ns with respect to the input signal. The maximum frequency of the input signal is 1.56 MHz and the minimum pulse width 320 ns, so it fulfills the requirements for this system.

For programming the delay, the delay line has a serial SPI interface or a parallel interface depending on the MD pin logic value of the delay line. In this design, the serial interface has been chosen (MD pin connected to ground) to minimize the FPGA ports used. The serial SPI interface digital delay pins have been connected directly to the FPGA since they have 3.3 V LVCMOS format. The serial SPI interface signals are the following,

- SI (*serial data input*) line: this is an input which will receive synchronously with the clock the data from the FPGA controller. Each data frame, will have 8 bits corresponding to the delay to be programmed. The data format will be big-endian (the most significant bit, MSB, first).
- SC (*serial data clock*) line: this is the clock input for receiving the data synchronously. The maximum frequency of the clock can be 80 MHz.
- AE (*address enable*) line: active high input in order to select the chip for a data transfer.

The digital delay line is biased with a 3.3 V digital supply level. The digital power supply is decoupled with three parallel capacitors (100 nF, 100 nF and 10 µF). The 100 nF capacitors are ceramic and 10 µF capacitor is a tantalum capacitor with low ESR.

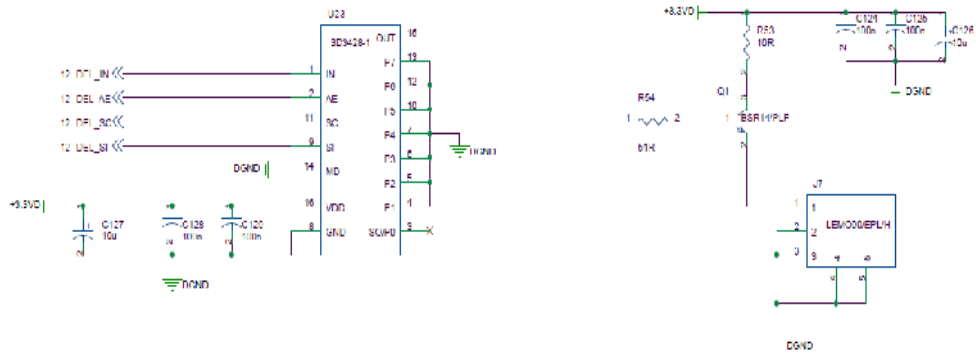

Figure 5.12. *Schematic circuit of the digital delay line and the 50 Ω driver.*

The output signal of the digital delay line is a 3.3 V LVCMOS signal which cannot drive a 50 Ω input since the maximum output current of the digital delay line is 15 mA. Therefore, a 50 Ω driver is necessary to drive a standard 50 Ω input of a pulse generator by means of a 50 Ω coaxial cable. Several alternatives have been considered for implementing this driver like a commercial 50 Ω driver. However, a simple driver based on a transistor [73] has been implemented. In the figure 5.12 the schematic circuit of the delay line and the 50 Ω driver can be seen. The transistor used is a standard NPN bipolar transistor, the BSR14 [74] from *Philips Semiconductors*. This is a general purpose bipolar transistor for switching and linear applications. The maximum collector current is 800 mA and the maximum collector-emitter voltage is 40 V. The transition frequency of this transistor is 300 MHz.

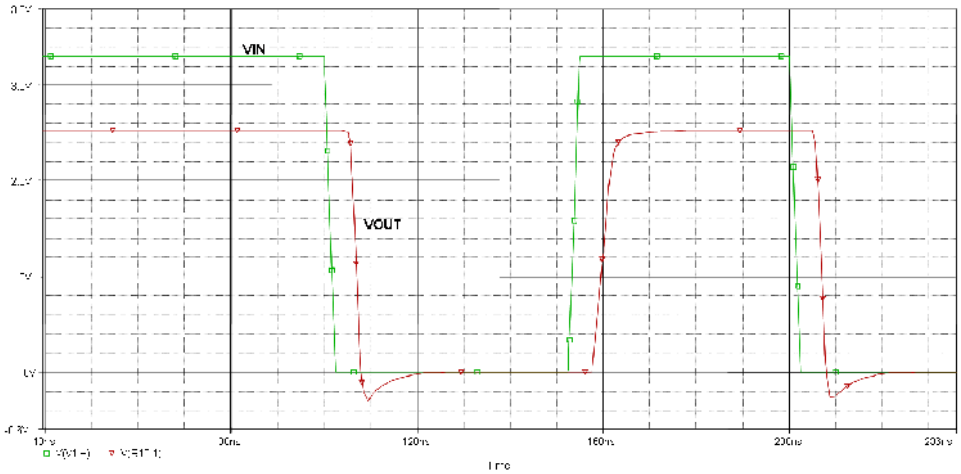

Figure 5.13. *PSpice simulation results of the 50 Ω driver considering a load of 50 Ω and a 50 Ω coaxial cable as the transmission line between the driver and the load.*

The circuit will work correctly when it is terminated with a 50 Ω resistor at the transistor emitter to ground (*i.e.* the circuit is an emitter follower). The collector resistor (10 Ω) and the base resistor (51 Ω) have been chosen in order to operate the transistor in the active region when the base signal is 3.3 V. When the base signal is 0 V the transistor is in the cut-off region. The collector resistor also limits the current on the 50 Ω load to have a 3.3 V LVCMOS format compatible voltage levels at the load. The driver is also biased with a 3.3 V digital supply level. The digital power supply is decoupled with three parallel capacitors (100 nF, 100 nF and 10 µF). The lower value capacitors are ceramic and the higher values capacitor is a tantalum capacitor with low ESR. In the figure 5.13 the *PSpice* simulation

results of this circuit are shown. It can be seen that the driver works correctly with a propagation delay of 6 ns approximately. The output signal of the driver is the TRIG OUT signal that will drive the laser source through a standard pulse generator. This signal is connected to a standard *Lemo* connector in order to drive the mentioned 50 Ω load by means of a 50 Ω coaxial cable.

5.2.5. Temperature converter

There is a NTC (*negative temperature coefficient*) thermistor at the daughter board as close as possible to the detectors and the Beetle chips in order to acquire temperature measurements as well. A specific temperature converter is used for digitizing the analogue signal corresponding to the voltage across the thermistor. In a NTC thermistor, the resistance value will decrease as long as the temperature increases. The relationship between the temperature and the thermistor's resistance is very nonlinear compared to a PTC (positive temperature coefficient) thermistor. However, a PTC thermistor is much more expensive than a NTC thermistor. Moreover, there are dedicated temperature converters for NTC thermistors and using these converters the measured temperature can be linear over a limited range.

The temperature converter used in this design is the MAX6682 [75] from *Maxim-Dallas*. The MAX6682 converts an external thermistor temperature dependent resistance directly into digital form. The thermistor and the external fixed resistor form a voltage divider that is driven by the temperature converter internal voltage reference. The MAX6682 measures the voltage across the external resistor, R_{EXT}, and produces a 10 bit plus sign digital output code dependent on that voltage by using an internal ADC.

Although the relationship between a thermistor's resistance and its temperature is very nonlinear, the voltage across R_{EXT} is reasonably linear over a temperature range, provided that R_{EXT} is chosen properly. The digital output is available as a 10-bit plus sign word. The relationship between the 11-bit digital word, D_{OUT}, and the voltage across R_{EXT} (normalized to V_R) is given by,

$$D_{OUT} = \frac{\left(\frac{V_{REXT}}{V_R} - 0.174387\right)}{0.010404} \quad (5.13)$$

where V_{REXT}/V_R is the voltage across R_{EXT} normalized to the value of V_R. The schematic circuit of the temperature converter is shown in the figure 5.14.

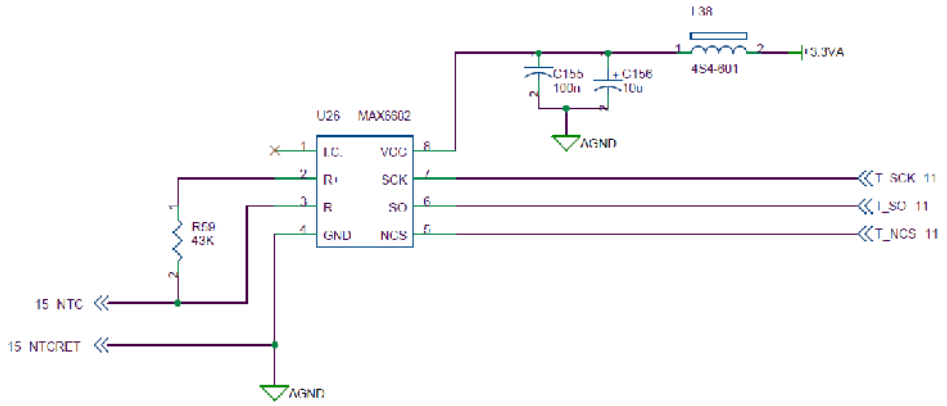

Figure 5.14. *Schematic circuit of the temperature converter.*

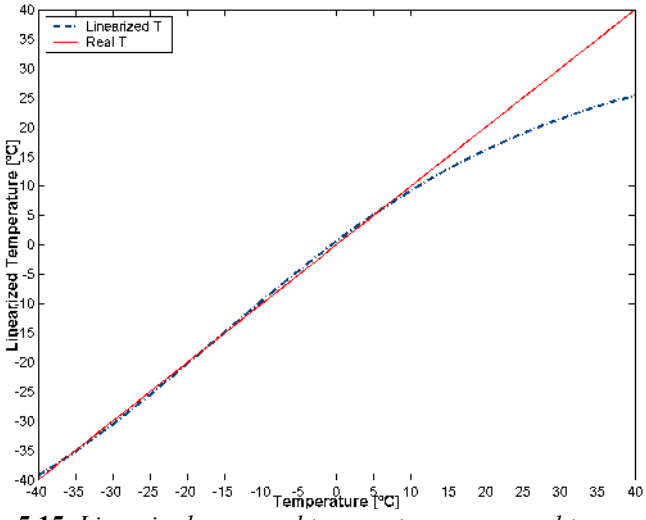

Figure 5.15. *Linearized measured temperature versus real temperature.*

The value of R_{EXT} has been fixed to 43 kΩ in order to have the maximum linearity at V_{REXT} over the temperature range of interest (-40 ºC to 10 ºC). Moreover, an equation adjusted experimentally will be used in the host computer software in order to have an almost complete linear response from -40 ºC to 10 ºC,

$$T[\,ºC\,] = 0.12 \cdot D_{OUT} - 39.8 \quad (5.14)$$

where T is the converter temperature readout in Celsius and D_{OUT}, the 11 bits digital converter readout. With this relationship, the temperature measured from the NTC at the daughter board over the range of interest is almost linear and the

deviation is lower than 1 ºC. This can be seen in the figure 5.15, where the converter readout temperature versus real temperature is shown, and in the figure 5.16, where the measurement deviation as the difference between converter readout and the real temperature versus real temperature is plotted.

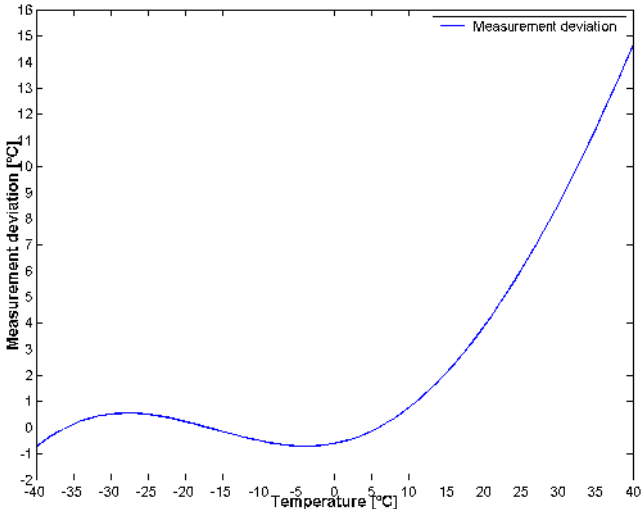

Figure 5.16. *Measurement deviation, as the difference between converter readout and the real temperature, versus real temperature.*

For reading out the temperature data of the temperature converter, the MAX6682 has a serial SPI slave interface. Therefore, a master controller has been developed and implemented in the FPGA in order to read out the. The temperature converter SPI is composed by three lines,

- SO (*serial data output*) line: this is an output which will send synchronously with the clock the temperature data to the FPGA controller. Each data frame, will have 11 bits corresponding to the temperature data. The data format will be big-endian (the most significant bit, MSB, first). The conversion time of the MAX6682 is 64 ms (*i.e.* each 64 ms there will be a new temperature date available to read it out).
- SCK (*serial data clock*) line: this is the clock input for reading out the data synchronously. The maximum frequency of the clock can be 10MHz.
- CS (*chip select*) line: active low input in order to select the chip for a data transfer.

All the serial digital signals are 3.3V LVCMOS compatible, so they are connected directly to the FPGA ports. The temperature converter is biased with 3.3 V digital supply level. This level is shared both by the analogue and digital interface of the MAX6682. However, the block has been treated as a fully analogue block. Therefore, the power supply input is bypassed with a combination of two parallel capacitors (100 nF and 10 μF) and a series ferrite bead (4S4 material with a DC resistance of 450 mΩ and a impedance of 600 Ω at 100 MHz) in order to maximize the noise immunity at high and low frequencies.

5.2.6. SDRAM

The amount of data that will be acquired both with the radioactive source setup and with the laser setup will be considerable. Although there is USB communication with the host computer which can achieve very fast data transfer rates, it is safer to store each acquisition in an on-system memory to manage correctly the data transfers.

The memory size must be big enough to store the acquired data while a standard acquisition is carried out. For instance, in a radioactive setup acquisition for each event (*i.e.* trigger input signal active meaning that a charged particle has crossed the silicon detector) will be acquired the following data,

- digitized Beetle chips analogue data: for each Beetle chip, a 16 bits data (10 bits data from the ADC plus six filling bits for obtaining a 16 bits word) will be acquired for each of the 128 channels. Therefore, there will be acquired 4096 bits (2 x 16 bits x 128 channels) corresponding to the Beetle chips analogue data.
- TDC measurement: a 32 bits time data will be acquired from the TDC for each event.
- Temperature converter data: a 16 bits data (11 bits from temperature converter readout plus 5 filling bits for obtaining a 16 bits word) will be acquired from the temperature converter.

Hence, for each radioactive source event 4144 bits (518 bytes) will be acquired. In case of laser setup, the same number of bits will be acquired for each event (*i.e.* an active pulse of TRIG OUT signal). Although there will not be TDC measurement since the system will have been synchronized, this empty space will be filled with 32 bits (all zero) in order to have symmetric data in both types of acquisitions. This facilitates the host computer software development (*i.e.* the same low level routines can be used in both cases).

The FPGA has on-chip RAM memory. However, its size is not enough for storing more than few tens of events when it would be interesting to store hundreds or thousands of events. Therefore, an external memory is necessary. In particular, a SDRAM memory could cope with the size requirement for thousands of events. The SDRAM memory has to be refreshed continuously since its storage cells are basically capacitors implemented with transistors which require being refreshed for maintaining the stored data. This fact involves a more complex controller. On the other hand, a SRAM (*static random access memory*) memory type has the advantage of an easier control but it has lower storage capability since its storage cells are flip-flops [76].

Having into account the previous considerations, an external SDRAM chip has been included in this system. The SDRAM used is the MT48LC32M8A2-7E [77] from *Micron Technology*. This is a 256 Mb SDR (*single data rate*) SDRAM distributed in four banks of 64 Mb. The digital interface of this SDRAM has the following signals,

- CLK: clock input signal. All SDRAM input signals are sampled on the positive edge of CLK. CLK also increments the internal burst counter and controls the output registers.
- CKE: clock enable input. CKE activates (high level) and deactivates (low level) the CLK signal. Deactivating the clock provides PRECHARGE power-down and SELF REFRESH operation (all banks idle), ACTIVE power-down (row active in any bank) or CLOCK SUSPEND operation (burst/access in progress). CKE is synchronous except after the device enters power-down and self refresh modes, where CKE becomes asynchronous until after exiting the same mode. The input buffers, including CLK, are disabled during power-down and self refresh modes, providing low standby power. CKE may be tied high.
- CS#: chip select input. CS# enables (registered low) and disables (registered high) the command decoder. All commands are masked when CS# is registered high, but READ/WRITE bursts already in progress will continue and DQM operation will retain its DQ mask capability while CS# is high level. CS# provides for external bank selection on systems with multiple banks. CS# is considered part of the command code.
- WE#, RAS#, CAS#: command inputs. They (along with CS#) define the command being entered depending on their logic levels.
- DQM: input/output mask. DQM is an input mask signal for write accesses and an output enable signal for read accesses. Input data is masked when DQM is sampled high during a WRITE cycle. The output buffers are placed in a High-Z state (two-clock latency) when DQM is sampled high

during a READ cycle.

- BA0-BA1: bank address inputs. BA0 and BA1 define to which bank the ACTIVE, READ, WRITE, or PRECHARGE command is being applied.
- A0-A12: address inputs. They are sampled during the ACTIVE command (row address A0-A12) and READ or WRITE command (column-address on A0-A9 with A10 defining auto precharge) to select one location out of the memory array in the respective bank. Signal A10 is sampled during a PRECHARGE command to determine if all banks are to be precharged (high level on A10) or one bank is selected (low level on A10). The address inputs also provide the operation code during a LOAD MODE REGISTER command.
- DQ0-DQ7: data input/output ports. Data bus for reading or writing from/to the SDRAM.

All these signals have 3.3 V LVCMOS format so they are connected directly to the FPGA ports. A FPGA controller manages the read/write access to the memory as well as the command control required by the SDRAM. The SDRAM supply ports are biased with a 3.3 V digital supply level. The digital power supplies are decoupled with three parallel capacitors (10 nF, 100 nF and 10 μF). The lower value capacitors are ceramic and the higher value capacitor is a tantalum capacitor with low ESR.

5.2.7. USB controller

The system has to communicate with the host computer software for sending the acquired data and status commands as well as for receiving the control commands required by the hardware part. This communication is carried out using a USB serial interface. The USB serial interface protocol, the electrical and the mechanical characteristics are detailed in an industry-standard specification [78]. The Universal Serial Bus was originally developed in 1995 by various industry leading companies. The major goal of USB was to define an external expansion bus which makes adding peripherals to a computer as easy as hooking up a telephone to a wall-jack. The driving goals were ease of use and low cost. These were enabled with an external expansion architecture which highlights,

- computer host controller hardware and software.
- Robust connectors and cable assemblies.
- Peripheral friendly master-slave protocols.
- Expandable through multi-port hubs.

The role of the system software is to provide a uniform view of IO (*input-output*) system for all applications software. It hides hardware implementation details so that application software is more portable. For the USB IO subsystem in particular, it manages the dynamic attach and detach of peripherals. This phase, called enumeration, involves communicating with the peripheral to discover the identity of a device driver that it should load, if not already loaded. A unique address is assigned to each peripheral during enumeration to be used for run-time data transfers. During run-time the host computer initiates transactions to specific peripherals, and each peripheral accepts its transactions and responds accordingly. Additionally the host computer software incorporates the peripheral into the system power management scheme and can manage overall system power without user interaction.

All USB peripherals are slaves that obey a defined protocol. They must react to request transactions sent from the host computer. The peripheral responds to control transactions that, for example, request detailed information about the device and its configuration. The peripheral sends and receives data to/from the host using a standard USB data format. This standardized data movement to/from the host computer and interpretation by the peripheral gives USB its enormous flexibility with little host computer software changes. USB 1.1 peripherals can operate at 12 Mb/s (full-speed) or 1.5 Mb/s (low-speed) while USB 2.0 can operate at 480 Mb/s (high-speed). The USB 2.0 is a backwards-compatible extension of the USB 1.1 specification and it uses the same cables, connectors and software interfaces. The new USB 3.0 specification [79] defines a maximum speed of 5 Gb/s and it is also backwards-compatible with USB 2.0.

Different options have been considered in order to implement a USB controller in the mother board. Finally a chip which can manage itself the USB protocol has been adopted as a solution since developing a SIE (*serial interface engine*) to manage the USB protocol is a very time-consuming task. Another requirement for this design was to use a controller which had its own host computer software driver for managing the communication.

The chip used in this system is the FT245R [80] from *FTDI*. This chip contains is a USB to parallel FIFO bidirectional data interface. The entire USB protocol is handled on the chip. Therefore, there is no need of a USB specific firmware programming as it is required with other chips which incorporate a microprocessor for this task [81, 82]. Furthermore, the manufacturer of this chip offers ready-to-use royalty-free USB drivers for different platforms (Windows, Linux, etc) in order to use them with the software application. This fact eliminates the requirement for USB driver development in most cases, *i.e.* the full bandwidth of the USB is not

required, as in this system. There are two types of USB drivers for this chip,

- VCP (*virtual COM port*) driver: this driver causes the FT245R to appear as an additional serial port available to the PC. Application software can access the FT245R in the same way as it would access a standard serial port. The data transfer rate with this driver is limited to 2.4 Mb/s.
- D2XX (*direct*) driver: this driver allows direct access to the FT245R through a DLL (*dynamic link library*). Application software can access the FT245R through a series of DLL function calls. The data transfer rate with this driver is limited to 8 Mb/s.

In this system, a VCP driver is used because the full USB bandwidth is not required since a SDRAM is used for temporally storing the acquired data. For this reason a 2.4 Mb/s data transfer rate is high enough for this application. Moreover, the software application development is easier with this driver since the USB port can be treated as a standard serial port.

The FT245R has two internal FIFO buffers. Data written in the device using the WR pin are stored in the 128 bytes FIFO transmit buffer. The USB host controller removes data from the FIFO transmit buffer by sending a USB request for data from the device data *In* endpoint. Data sent from the USB host controller to the FIFO via the USB data *Out* endpoint is stored in the 256 bytes FIFO receive buffer and is removed from this buffer by reading the contents of the FIFO using the RD# pin. For the communication with the system FPGA, the FT245R has a 3.3 V LVCMOS digital interface with the following signals,

- D0-D7: data input/output bus. This bus is used to read/write data from/to the FT245R internal FIFO buffers.
- PWEN#: power enable output. It goes low after the device is configured by USB, then high during USB suspend.
- RD#: active low read control input. The current FIFO data byte on D0-D7 bus is enabled when RD# is low. Fetched the next FIFO data byte (if it is available) from the receive FIFO buffer when RD# goes from high to low level.
- WR: active high write control input. The data byte on the D0-D7 pins is written into the transmit FIFO buffer when WR goes from high to low.
- TXE#: active low transmit enable output. When it is high, data into the FIFO cannot be written. When it is low, data can be written into the FIFO by strobing WR high, then low. During reset this signal pin is tri-state, but pulled up to VCCIO via an internal 200kΩ resistor.
- RXF#: active low receive enable output. When it is high, data from the

FIFO cannot be read. When low, there is data available in the FIFO which can be read by strobing RD# low, then high again. During reset this signal pin is tri-state, but pulled up to VCCIO via an internal 200kΩ resistor.

These signals have been connected directly to the FPGA ports since a custom controller interfaces the FT245R chip. The FT245R has its own clock generator so there is no need to supply an external clock to the chip. This chip has also an USB transceiver cell which provides the USB 1.1/USB 2.0 full-speed physical interface to the USB cable. The output drivers provide 3.3V level slew rate control signaling, whilst a differential receiver and two single ended receivers provide USB data in, SEO (*single-ended zero*) and USB *Reset* condition detection. This transceiver also incorporates internal USB series resistors on the USB data lines (USBDP pin and USBDN pin), and a 1.5kΩ pull up resistor on USBDP.

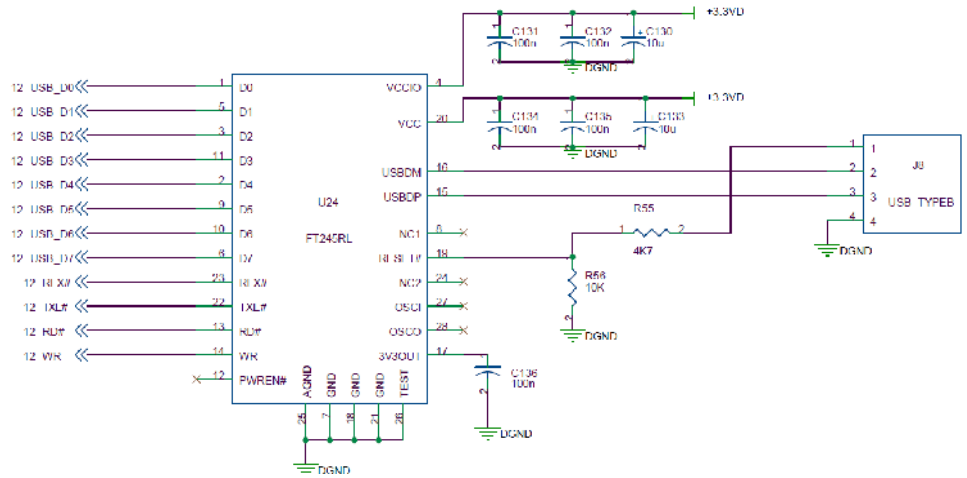

Figure 5.17. *Schematic circuit of the FT245R implementation in the mother board.*

There is also a SIE block which performs the parallel to serial and serial to parallel conversion of the USB data, in accordance to the USB 2.0 specification. The USB Protocol Engine block of the FT245R manages the data stream from the device USB control endpoint. It handles the low level USB protocol requests generated by the USB host controller and the commands for controlling the functional parameters of the FIFO. The integrated *Reset Generator* block provides a reliable power-on reset to the device internal circuitry on power up. A RESET# input is provided to allow other devices to reset the FT245R. Finally, the internal EEPROM in the FT245R can be used to store USB *Vendor ID* (VID), P*roduct ID* (PID), device serial number, product description string, and various other USB configuration descriptors. The device is supplied with the internal EEPROM

settings already programmed.

In figure 5.17, a schematic circuit of the FT245R USB controller implemented in this system is shown. The USB data lines (USBDM and USBDP) are connected directly to the corresponding pins at the USB connector. The RESET# port is connected through a resistor divider to the USB power supply pin at the UDB connector. Since the chip is not biased via the USB power supply line but it is biased by means of the system 3.3 V digital supply, the RESET# pin must be connected this way in order to keep the FT245R chip on a reset state while there is no USB connection. The 3.3 V digital supply levels are decoupled with three parallel capacitors (100 nF, 100 nF and 10 μF). The 100 nF capacitors are ceramic and the 10 μF capacitor is a tantalum capacitor with low ESR.

5.2.8. FPGA hardware and associated circuits

There is a FPGA in the system which implements the required logic for controlling and interfacing the different hardware blocks. A FPGA is a semiconductor device containing programmable logic components, so-called logic blocks, and programmable interconnects [83-86]. Logic blocks can be programmed to perform the function of basic logic gates or more complex combinational functions such as decoders or simple mathematical functions. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. Moreover, a FPGA will contain flexible input/output structures which have various flip-flops and can implement different digital signal formats (such as LVCMOS, LVDS or LVTTL) as well as different input/output paths.

The origin of FPGAs is in *complex programmable logic devices* (CPLDs) of the early to mid 1980s. CPLDs and FPGAs include a relatively large number of programmable logic elements. CPLD logic gate densities range from the equivalent of several thousand to tens of thousands of logic gates, while FPGAs typically range from tens of thousands to several million. The main differences between CPLDs and FPGAs are based on the architecture. A CPLD has a more restrictive structure consisting of one or more programmable sum-of-products logic arrays feeding a relatively small number of clocked registers. As a result, it has less flexibility, with the advantage of more predictable timing delays and a higher logic-to-interconnect ratio. On the contrary, the FPGA architectures are dominated by interconnect. This makes them more flexible, in terms of the range of designs that are practical for implementation within them, but also more complex for implementing a design with them. Another notable difference between CPLDs and FPGAs is the presence in most FPGAs of higher-level embedded functions (such as

adders and multipliers) and embedded memories. Furthermore, a more recent trend has been to combine custom logic with pre-designed IP (*Intellectual Property*) cores, such as software embedded microprocessors and related peripherals, to form a complete system on a programmable chip.

The different types of commercial FPGAs differ basically in the configuration technology, the logic block architecture and the interconnect architecture. FPGAs are configured by means of electrically programmable switches. There are three main technologies for these switches. One of them is based on CMOS static RAM (SRAM) memory cells. With this technology the FPGA is in-system programmable and re-programmable but requires an external boot device like an EEPROM (*Electrically Erasable Programmable Read-Only Memory*). Another one is the CMOS anti-fuse technology. With this technology the FPGA is one time programmable and it does not require an external boot device. The third one is based on flash EEPROM memory cells. With this technology the FPGA is in-system programmable, re-programmable and it does not require an external boot device. Regarding the logic block architecture, the two main architectures are the logic blocks based on multiplexors and the logic blocks based on LUTs (*Look-Up Tables*) which are implemented with SRAM memories. Both the multiplexors and the LUTs can implement combinational functions which are combined with memory blocks in the logic cells.

In order to define the behaviour of the FPGA (*i.e.* the logic which will be implemented in the FPGA) the user must provide a hardware description language (HDL) or a schematic design. The most common HDLs are VHDL [87] and Verilog [88]. Then, using an electronic design automation tool, a technology-mapped netlist is generated. The netlist can then be fitted to the actual FPGA architecture using a process called *place-and-route*, usually performed by the FPGA company proprietary *place-and-route* software. The user will validate the map, place and route results via timing analysis, simulation, and other verification methodologies. Once the design and validation process is complete, the binary file generated (also using the FPGA company proprietary software) is used to configure the FPGA, which is transferred to the FPGA via a serial interface (JTAG) interface or to external memory device like an EEPROM.

The FPGA used in this system is the Spartan-3 XC3S400-PQ208 [89] from *Xilinx*. This FPGA has 400,000 system gates so is a low-medium density device. The Spartan-3 family is composed of FPGAs based on SRAM configuration technology. The Spartan-3 family architecture consists of five fundamental programmable functional elements,

- CLBs (*Configurable Logic Blocks*) which contain SRAM LUTs to implement logic and storage elements that can be used as flip-flops or latches. CLBs can be programmed to perform a wide variety of logical functions as well as to store data.
- IOBs (*Input/Output blocks*) of these FPGAs support bidirectional data flow plus three-state operation. Twenty-four different signal standards are available. *Double Data-Rate* (DDR) registers are included. The *Digitally Controlled Impedance* (DCI) feature provides automatic on-chip terminations.
- Block RAM provides data storage in the form of 18-Kbit dual-port blocks.
- Multiplier blocks accept two 18-bit binary numbers as inputs and calculate the product.
- *Digital Clock Manager* (DCM) blocks provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase shifting clock signals.

These elements are organized as shown in figure 5.18. A ring of IOBs surrounds a regular array of CLBs. The XC3S400 device has two RAM columns. Each column is made up of several 18-Kbit RAM blocks, each block is associated with a dedicated multiplier. The DCMs are positioned at the ends of the outer block RAM columns. The Spartan-3 family features a network of traces and switches that interconnect all five functional elements, transmitting signals among them. Each functional element has an associated switch matrix that permits multiple connections to the routing.

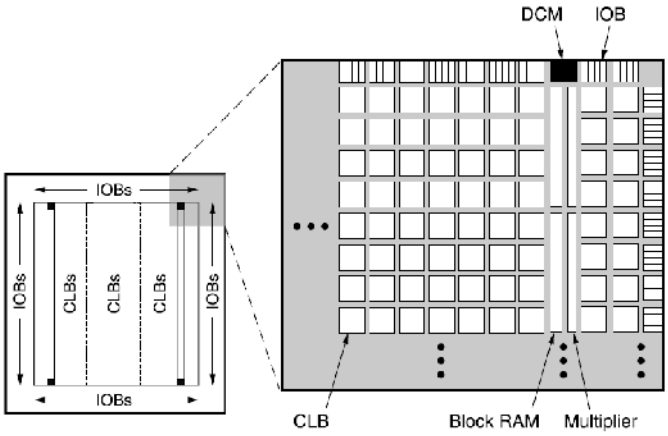

Figure 5.18. *Spartan-3 family architecture. Figure taken from [89].*

The XC3S400-PQ208 has a plastic quad flat package with 208 pins. For the

XC3S400-PQ208, IOBs are allocated among eight banks, so that each side of the device has two banks. Each bank has two dedicated V_{CCO} lines, *i.e.* output voltage supply per bank. There are also four internal core voltage supply pins, V_{CCINT} (1.2 V), and eight auxiliary voltage supply pins, V_{CCAUX} (2.5 V). There are 28 ground (GND) lines distributed uniformly along the IOBs. Moreover, there are eight special IOBs that can be used as global clock buffer inputs, GCLKs, two in each of the upper and bottom banks (Bank 0, Bank 1, Bank 4 and Bank 5). Finally, there are eleven dedicated pins for configuration and JTAG communication as well as twelve dual pins that are used for configuration and, then, as standard inputs/outputs. The rest of the pins, 121 pins, can be used as standard inputs/outputs (some of them as differential input/output pairs). The table 5.1 summarizes the FPGA ports used in this system.

Bank 7 (from top to bottom): V_{CCO_7} = 3.3 V					
Signal name	**Hardware block**	**Pin number**	**Signal type**	**I/O**	**Signal standard**
DQ1-DQ0	SDRAM	2-3	Single-ended	Bidirectional	3.3 V LVCMOS
Trigger	Fast Control	4		Output	
CLK		5			
Testpulse		7			
Reset		9			
SDA	Slow Control	10		Bidirectional	
SCL		11			
SCK	Temperature	12		Output	
SO		13		Input	
NCS		15		Output	
DataValid 0	Fast Control	16		Input	
DataValid 1		18			
D0-D5	ADC0	20- 37			
Bank 6 (from top to bottom): V_{CCO_6} = 3.3 V					
Signal name	**Hardware block**	**Pin number**	**Signal type**	**I/O**	**Signal standard**
D6-D9	ADC0	28-33	Single-ended	Input	3.3 V LVCMOS
PD		34		Output	
SCLK		35			
D0-D9	ADC1	37-50		Input	
PD		51		Output	

SCLK		52			
Bank 5 (from left to right): V_{CCO_5}= 3.3 V					
Signal name	**Hardware block**	**Pin number**	**Signal type**	**I/O**	**Signal standard**
M1_M0	Configuration	54-56	Single-ended	Input	2.5 V LVCMOS
SDI	DAC	61		Output	3.3 V LVCMOS
CLK		62			
NCS		63			
NLOADDACS		64			
NRESET		65			
Bank 4 (from left to right): V_{CCO_4} = 2.5 V					
Signal name	**Hardware block**	**Pin number**	**Signal type**	**I/O**	**Signal standard**
SIN1	Trigger Cond.	79-80	Differential	Input	2.5 V LVPECL
INIT_B	Configuration	83	Single-ended	Bidirectional	2.5 V LVCMOS
SIN2	Trigger Cond.	86-87	Differential	Input	2.5 V LVPECL
DIN	Configuration	92	Single-ended		2.5 V LVCMOS
PNEG	Trigger Cond.	94-95	Differential		2.5 V LVPECL
PPOS		100-101			
DONE	Configuration	103	Single-ended	Bidirectional	2.5 V LVCMOS
CCLK		104			
Bank 3 (from bottom to top): V_{CCO_3} = 3.3 V					
Signal name	**Hardware block**	**Pin number**	**Signal type**	**I/O**	**Signal standard**
CLK_REF	TDC	106	Single-ended	Output	3.3 V LVCMOS
RST_N		107			
ENSTOP1		108			
STOP1		109			
D0-D7		111-120		Bidirectional	
START		122		Output	
INT_FLAG		123		Input	
CSN		124		Output	
WRN		125			
RDN		126			

ALE		128			
TXE#	USB	130		Input	
RFX#		131			

Bank 2 (from bottom to top): V_{CCO_2} = 3.3 V

Signal name	Hardware block	Pin number	Signal type	I/O	Signal standard
WR	USB	132	Single-ended	Output	3.3 V LVCMOS
RD#		133			
D0-D7		135-144		Bidirectional	
SI	Delay Line	146		Output	
SC		147			
AE		148			
IN		149			
nPWEN	USB	152		Input	
LED_RED	LEDs	154		Output	
LED_GREEN		155			
NRESET	System Reset	156		Input	

Bank 1 (from right to left): V_{CCO_1} = 3.3 V

Signal name	Hardware block	Pin number	Signal type	I/O	Signal standard
TDO	Configuration	158	Single-ended	Output	2.5 V LVCMOS
TCK		159		Input	
TMS		160			
DQ7-DQ4	SDRAM	161-166	Single-ended	Bidirectional	3.3 V LVCMOS
DQM		167			
CKE		168		Output	
A12-A11		169-171			
A9-A6		172-178			
OSC_IN	System Clock	181		Input	
A5	SDRAM	182		Output	

Bank 0 (from right to left): V_{CCO_0} = 3.3 V

Signal name	Hardware block	Pin number	Signal type	I/O	Signal standard
A4	SDRAM	183	Single-ended	Output	3.3 V LVCMOS

CLK_FB		184		Input	
CLK		185		Output	
A3-A0		187-191			
A10		194			
BA1-BA0		196-197			
CS#		198			
RAS#		199			
CAS#		200			
WE#		203			
DQ3-DQ2		204-205		Bidirectional	
HSWAP_EN	Configuration	206		Input	2.5 V LVCMOS
PROG_B		207		Bidirectional	
TDI		208		Input	

Table 5.1. *Summary of the FPGA ports used in the system.*

Each voltage supply level has one purpose. The V_{CCO} voltage supplies bias the IOBs of the corresponding bank. Therefore, in the same bank inputs/outputs with compatible formats should be used (*i.e.* with the same V_{CCO} level). In this design just two different digital V_{CCO} levels are used, 2.5 V for the bank 4 and 3.3 V for the rest of banks. The core supply digital level V_{CCINT} is 1.2 V and it biases the internal logic of the FPGA. Finally, the auxiliary supply voltage level, V_{CCAUX}, is 2.5 V and it biases the dedicated configuration IOBs.

The bypass and decoupling capacitors of the FPGA are of vital importance since the FPGA is probably the main source of noise of the mother board. Two main reasons are responsible for this fact. First, the FPGA is the most demanding IC of the mother board in terms of power. Second, the transient current demands are higher in the FPGA than in other mother board ICs due to its nature. Therefore, the number of capacitors, their values and type as well as their situation regarding the FPGA power pins are issues to be considered [90].

Concerning the number of capacitors, the basic objective is to have at least one capacitor per V_{CC} pin used on the device. Therefore, the effective number of V_{CC} pins for each supply must be determined. All supplies must be considered: V_{CCINT}, V_{CCAUX} and V_{CCO}. V_{CCAUX} and V_{CCINT} pins must always be fully decoupled, that is, they must always have one capacitor per pin. V_{CCO} can be prorated according to I/O utilization. The number of V_{CCO} pins used by a device can be determined based on the *simultaneously switching output* (SSO) restrictions given in the device

documentation. The SSO describes the maximum number of user output pins, of a given output signal standard, that should simultaneously switch in the same direction, while maintaining a safe level of switching noise. The summary of the SSO calculation for the system FPGA is shown in table 5.2. In this calculation, configuration I/O ports have not been computed since they are used only during configuration and bidirectional ports have been computed as outputs.

For this system, as can be seen in table 5.2, considering that all the bidirectional pins will act as outputs at the same time and that all the outputs will switch at the same time, the SSO limit is maintained if all V_{CCO} pins are used from bank 0 to bank 7 (*i.e.* two supply pins per bank are considered). Therefore, four 1.2 V power supply pins (V_{CCINT}), ten 2.5 V power supply pins (V_{CCAUX} pins and V_{CCO4} pins) and twelve 3.3 V power supply pins (V_{CCO} pins from bank 0 to bank 3 and V_{CCO} pins from bank 5 to bank 7) are considered. As a result, a minimum of 12 capacitors will be needed for the 3.3 V supply levels, 10 capacitors for the 2.5 V supply levels and 4 capacitors for the 1.2 V supply levels.

Bank	Voltage	Number of I/O	Number of outputs	Standard used for outputs	SSO Limit
Bank 0	3.3 V	17	16	LVCMOS33 6 mA (Slow)	16
Bank 1		14	13		
Bank 2		18	16		
Bank 3		20	17		
Bank 4	2.5 V	8	0	LVPECL25	6
Bank 5	3.3 V	5	5	LVCMOS33 6 mA (Slow)	16
Bank 6		18	4		
Bank 7		19	11		

Table 5.2. *Summary of the SSO calculation for the system FPGA. Configuration I/O ports have not been computed since they are used only during configuration. Bidirectional ports have been computed as outputs.*

Given the number of discrete capacitors needed as determined above, a distribution of capacitor values adding up to that total number must be determined. To cover a broad range of frequencies, a broad range of capacitor values must be used. The proportion of high-frequency capacitors to low-frequency capacitors is an important factor. The objective of a parallel combination of a number of values of capacitors is to keep low and flat power supply impedance over frequencies from the 500 kHz range to the 500 MHz range.

Both large value (low frequency) and small value (high frequency) capacitors

are needed. Small value capacitors tend to have less of an impact on the total impedance profile, so a greater number of small value capacitors are needed to yield the same impedance level impact as a small number of large value capacitors. To keep the impedance profile smooth and free of anti-resonance spikes, a capacitor is generally needed at least in every decade of the capacitor value range. The typical ceramic capacitor range generally spans values from 1 nF to 4.7 µF. The exact value of these capacitors is not critical. What is critical is having some capacitor value in every order of magnitude over this range. It is better to have as many values as possible, as a flatter impedance profile is yielded.

A ratio of capacitors giving relatively flat impedance is one where the quantity of capacitors is roughly doubled for every decade of decrease in size. In addition, low-frequency capacitance in the form of tantalum or electrolytic capacitors is needed. These large capacitors typically have a higher ESR than ceramic chip capacitors, making them less likely to contribute to anti-resonance spikes. For this reason, it is not necessary to maintain the rule of one value per decade. Generally, one value in the 470 µF to 1000 µF range is sufficient. Following these guidelines the following capacitors have been selected for the different power supply levels, as can be seen in table 5.3. The placement of these capacitors is detailed in section 5.2.10.

The FPGA is clocked with an externally generated clock signal with a frequency of 40 MHz. This clock signal is generated by a SMD (s*urface mount device*) oscillator. The clock signal is 3.3 V LVCMOS format compatible and it is connected to a special general clock buffer input of the FPGA. The oscillator is biased with a digital 3.3V supply level which is decoupled with a 100 nF capacitor.

Supply level	680 µF (4%)	2.2 µF (14%)	100 nF (27%)	10 nF (55%)	Calculated quantity	Final quantity
1.2 V	1 (0.16)	1 (0.56)	1 (1.08)	2 (2.2)	4	5
2.5 V	1 (0.4)	2 (1.4)	3 (2.7)	6 (5.5)	10	14
3.3 V	1 (0.48)	2 (1,68)	4 (3.24)	7 (6.6)	12	14

Table 5.3. *Summary of the bypassing and decoupling capacitors of the FPGA. In the first row between brackets, the optimum percentage of the calculated quantity for each value of capacitors is established [90]. Between the brackets of the rest of rows, the calculated quantity according to the optimum percentage for each capacitor value is reflected. The final quantity column informs about the final number of capacitors required for each supply level.*

There have been included to LEDs in order to inform to the user about the

system status, one is green and the other one is red. Each LED is controlled from a FPGA signal. However, for each LED a switching circuit has been implemented using a general purpose bipolar transistor [74] for protecting the FPGA outputs. The collector and base resistors of both circuits have been calculated for operating the transistor in saturation and cut-off mode depending on the base input signals (3.3 V or 0 V, respectively). Moreover, the recommended forward current of each LED have been considered as well. Both circuits have been biased with digital 3.3 V decoupled supply levels.

An external reset circuit has been implemented for having a manual system reset. The circuit consists of a pushing button switch biased with a decoupled 3.3 V digital level trough a current limiting resistor. When the button switch is pushed the output of the circuit is connected to ground, otherwise the output of the circuit has a 3.3 V voltage level. Obviously, the output of this circuit is connected directly to a FPGA port. The possible switch bounces are filtered by the FPGA logic so there is no need of an external hardware filter.

Finally, an external flash PROM (p*rogrammable read only memory*) has been included for FPGA configuration. The flash PROM used is the XCF02S [91] from *Xilinx*. This device has a capacity of 2 Mb and it is the device recommended for the FPGA used in this system. The PROM is programmed from a PC by means of JTAG protocol (*joint test action group*, standard IEEE 1149.1) using a dedicated software application. Therefore, a specific connector has been included in the motherboard for this purpose. The JTAG signals have been also connected to the JTAG ports of the FPGA so the FPGA can be accessed directly via this connector. On the other hand the PROM has been connected to the FPGA configuration ports so that the configuration data which have been programmed on the PROM can be transferred to the FPGA in serial mode. With this connection scheme, the FPGA acts as a master and the PROM as a slave. When the system is powered on, the FPGA automatically loads the data in the PROM (if any) generating the serial clock in the CCLK pin. This connection scheme is specified both in [89] and [91].

5.2.9. Supply system

The supply system of the whole hardware has been included in the mother board. First of all, the supply current and voltage requirements of the system were calculated considering the current and voltage requirements of each part of the hardware in the maximum possible consumption. These requirements are summarized in table 5.4.

Analogue power		Digital Power				DB Power	
		FPGA		Total			
5 V	204mA	1.2 V (V_{CCINT})	1210 mA	1.2 V	1210 mA	5 V	500 mA
		2.5 V (V_{CCAUX})	71 mA				
-5 V	221 mA	2.5 V (V_{CCO1})	66 mA	2.5 V	342 mA		
		3.3 V (V_{CCO2})	222 mA				
		Other					
3.3 V	104 mA	2.5 V	205 mA	3.3 V	914 mA		
		3.3 V	692 mA				
Analogue power		**Digital Power**				**DB Power**	
5 V ± 5%	245 mA	1.2 V ± 5%		1500 mA		5 V ± 5%	600 mA
-5 V ± 5%	266 mA	2.5 V ± 5%		411 mA			
3.3 V ± 5%	125 mA	3.3 V ± 5%		1097 mA			

Table 5.4. *Summary of the calculated power supply requirements. The calculation has been divided by voltage levels and by analogue circuits, digital circuits and daughter board circuits. The final currents have been increased with a 20 % of margin.*

Once the system supply requirements were calculated, the system supply design was tackled. A parallel supply system has been designed from a common input regulated *DC* voltage of 5 V. This input voltage level is generated from *AC* mains with a desktop *AC-DC* adapter. As a first approximation, an *AC-DC* adapter with a supply current up to 5 A has been considered. The output of *AC-DC* desktop adapter is connected directly to the mother board through a 2.1 mm *DC* power connector. Close to the *DC* power connector, an integrated EMI (*electromagnetic interference*) filter has been placed in order to reject the high frequency input noise as well as to reduce the input ripple. The selected filter is the BNX022 [92] from *Murata*. The equivalent circuit of the filter consists in LC filtering stages for both the differential mode noise and the common-mode noise. The insertion loss characteristics of this filter can be seen in figure 5.19. The insertion loss for each frequency is the rate in decibels between the voltage output and the voltage input of the filter.

Once the common input *DC* voltage has been filtered, it is connected to the different inputs of *DC-DC* converters in order to generate the required *DC* supply levels at the required currents, having a parallel scheme. A 5 V supply level with a

current up to 600 mA must be generated for the daughterboard. This 5 V supply level must be isolated since it is sent to the daughter board through flat ribbon cable in order to bias the Beetle chips and other hardware of the daughter board (see chapter four for more details). The TEL3-0511NP isolated *DC-DC* converter [93] from *Traco Power* is used. The input voltage range of this *DC-DC* converter is from 4.5 V to 9 V and the output is 5 V with a maximum current of 600 mA. The load voltage regulation (change in the output voltage over the specified change in the output load) of this *DC-DC* converter over a 10-100% of load variation is ±0.5% of the output voltage while the line voltage regulation (change in the output voltage for a given change in the input voltage) is ±0.5% of the output voltage. Its switching frequency is 300 kHz with an output ripple and noise over a 20 MHz bandwidth is below 60 mV peak-to-peak.

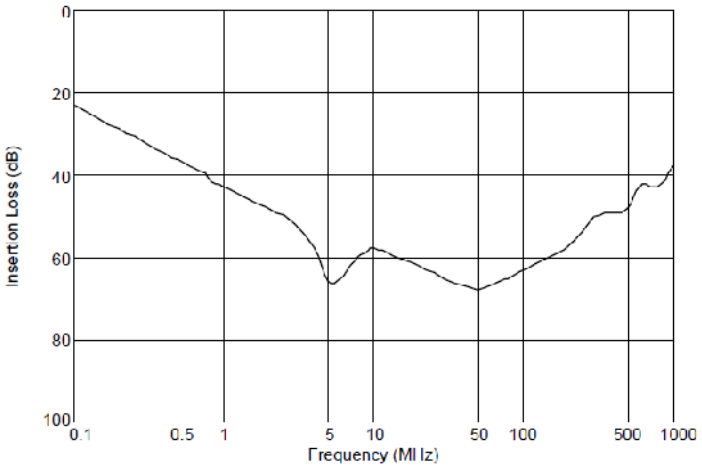

Figure 5.19. *Insertion loss characteristics of the power supply EMI filter. Figure from [92].*

For the TEL3-0511NP two filters have been implemented, one at the input and other one at the output, as can be seen in figure 5.20. The input filter on a switching power supply has two primary functions [94]. One is to prevent EMI, generated by the switching source, from reaching the power line and affecting other equipment. The second is to prevent high-frequency voltage on the power line from passing through the output of the power supply. The output filter on a switching power supply has the main purpose of reducing the output ripple and noise generated by the switching source [95] and to avoid that the high-frequency noise on the output power bus of the *DC-DC* converter going trough the *DC-DC* converter to the input [96].

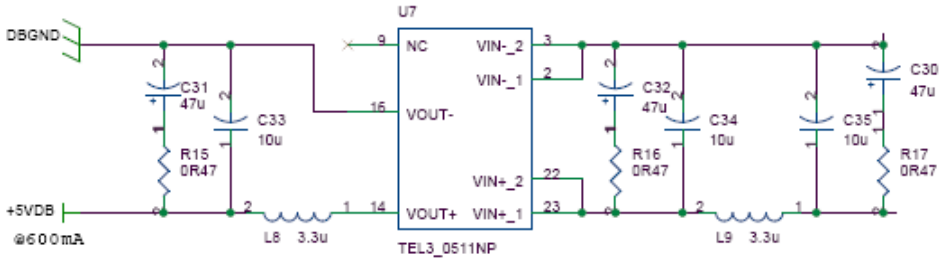

Figure 5.20. *Schematic circuit of the daughterboard isolated DC-DC converter and filters.*

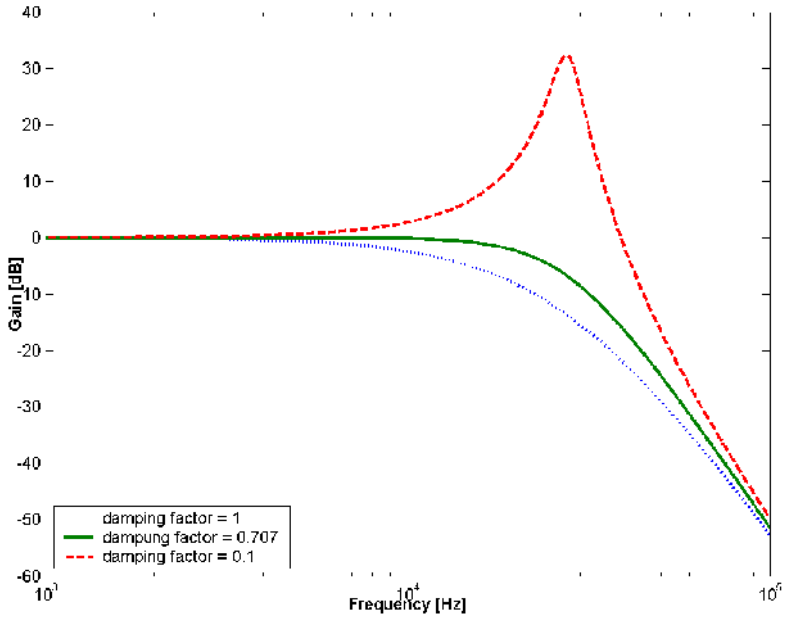

Figure 5.21. *Gain response versus frequency of a LC filter with a cut-off frequency (f_c) of 27.7 kHz for three different damping factors (ζ=0.1, ζ =0.707 and ζ =1).*

The input filter for the TEL3-0511NP is a second order damped pi-filter. It is composed by a series inductor and two capacitors of equal value in parallel. The cut-off frequency, f_c, of this filter is given by

$$f_c = \frac{1}{2 \cdot \pi \cdot \sqrt{L \cdot C}} \quad (5.15)$$

where L is the value of the inductor and C the value of the parallel capacitors. In this case the cut-off frequency of the filter has been fixed to 27.7 kHz by choosing the appropriate values for L and C (*L9*, *C34* and *C35* in figure 5.20 for the input

filter), around one decade below the switching frequency of the *DC-DC* converter (300 kHz). The response of this filter is of second order, that is, the attenuation after the cut-off frequency is -40 dB/decade. One of the critical factors involved in designing a second order filter is the attenuation characteristics at the corner frequency f_c. The gain near the cut-off frequency could be very large, and amplify the noise at that frequency as can be seen in figure 5.21.

The damping factor, ζ,, describes the gain at the corner frequency. For a damping factor higher than one, the two poles are complex and the imaginary part gives the peak behaviour at the resonant frequency. As the damping factor becomes smaller, the gain at the corner frequency becomes larger. With a damping factor equal to one the imaginary component is null and there is no peaking. A poor damping factor on the filter design could have other side effects on the final performance of the system. It can influence the transfer function of the *dc-dc* converter and cause some oscillations at the output of the converter. On the design point of view, a good compromise between size of the filter and performance is obtained with a minimum damping factor of $1/\sqrt{2}$ (0.707), which provides -3 dB attenuation at the corner frequency and stability.

Therefore, a damping resistor R_d (R16 and R17 in figure 5.20 for the input filter) in series with a capacitor C_d (C30 and C32 in figure 5.20 for the input filter) has been introduced all in parallel with the filter capacitors. The purpose of resistor R_d is to reduce the output peak impedance of the filter at the cut-off frequency. The capacitor C_d blocks the *dc* component of the input voltage avoiding the power dissipation on R_d. The capacitor C_d should have lower impedance than R_d at the resonant frequency as well as it should have a bigger value than the filter capacitor in order to not affect the cut-off point of the main filter. Considering that C_d is four times bigger than the filter capacitor C, the optimum value of the damping resistor for a damping factor of $1/\sqrt{2}$ (0.707) is given by,

$$R_d = \sqrt{\frac{L}{C}} \quad (5.16)$$

where L is the value of the filter inductor. On the other hand, capacitors with low ESL and low ESR must be selected for high frequency attenuation and for ripple current capability, respectively. Hence, the filter capacitors are ceramic capacitors and the *DC* blocking capacitors are tantalum capacitors. Filter inductors should be designed to reduce parasitic capacitance as much as possible. The *DC* current rating of the inductor also needs consideration, a rating approximately twice the supply current is recommended, as well as the *DC* resistance and the self-resonance frequency (it must be quite higher than the operating frequency). Therefore, a low

inductance has been selected (3.3 μH) with a rated current of 5.5 A (for an input *DC-DC* converter full load expected current of 820 mA), a *DC* resistance of 12 mΩ and a self-resonance frequency of 50 MHz.

The output filter for the TEL3-0511NP is a second order damped LC filter. An LC filter has been chosen for the output since it is not expected that the daughter board hardware biased with this supply level is going to generate a considerable noise level. Therefore, the aim is to reduce the output ripple and noise generated by the switching *DC-DC* converter. The behaviour of this output filter is governed by equations 5.15 and 5.16 as in the case of the input filter. The selection of the components has been carried out following the same guidelines and they are the same as well.

For generating the analogue supply levels for the motherboard (5 V, -5 V and 3.3 V) an isolated *DC-DC* converter and a LDO regulator have been used. The isolated *DC-DC* converter is used to protect the analogue hardware against the digital hardware of the mother board. This converter is the NMXSOD0505SOC [97] from *C&D Technologies*. Its input voltage range is from 4.5 V to 5.5 V and it has dual output (±5 V) with a maximum current of 500 mA per output. The load voltage regulation over a 10-100 % of load variation is 0.5% of the output voltage, while the line voltage regulation is ±0.5% of the output voltage. Its switching frequency is 70 kHz with an output ripple and noise over a 20 MHz bandwidth below 150 mV peak-to-peak.

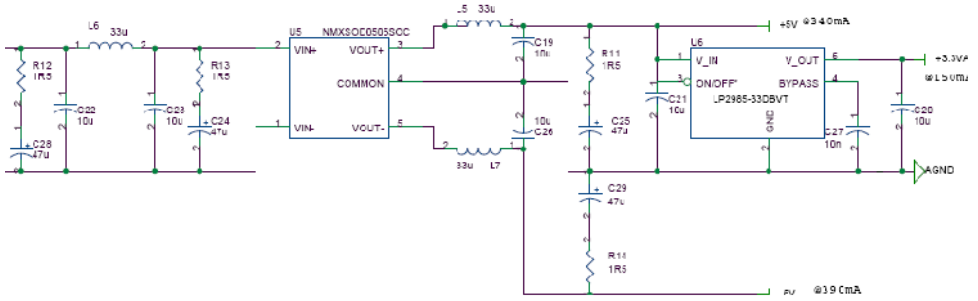

Figure 5.22. *Schematic circuit of the analogue isolated DC-DC converter, regulator and filters.*

Three filters, one at the input and one for each output, have been implemented for this converter as shown in figure 5.22. The input filter for the NMXSOD0505SOC is a second order damped pi-filter. The cut-off frequency of the filter has been fixed to 8.7 kHz by choosing the appropriate values for *L* and *C* (L6, C22 and C23 in figure 5.22), around one decade below the switching frequency of the converter (70 kHz). The output filters are second order damped

LC filters. The cut-off frequency of the filters has been fixed to 8.7 kHz as well. LC filters have been chosen for the output since it is not expected that the analogue hardware biased with this supply level is going to generate a considerable noise level. Therefore, the aim is to reduce the output ripple and noise generated by the switching *DC-DC* converter. The filter capacitors are ceramic capacitors and the *DC* blocking capacitors are tantalum capacitors. Inductances with low values have been selected (33 μH) with a rated current of 2.1 A (for an input expected current of 1200 mA), a *DC* resistance of 75 mΩ and a self-resonance frequency of 10 MHz.

From the 5 V output of the NMXSOD0505SOC, a 3.3 V analogue supply level is generated by means of the LP2985 [54] LDO linear regulator. This regulator is also used at the daughter board and it was presented in chapter four with the LDO linear regulator operation principles. It is a LDO linear regulator with a typical dropout of 280 mV at 150 mA load current. The capacitors at the input and output of the regulator are placed for the regulator stability and for noise minimization, and they have the recommended values from the manufacturer.

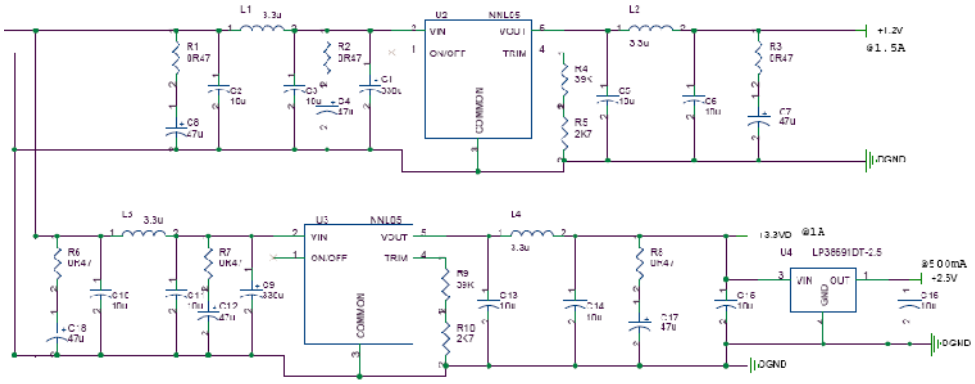

Figure 5.23. *Schematic circuit of the digital non-isolated dc-dc converters, regulator and filters.*

The digital supply levels (1.2 V, 2.5 V and 3.3 V) are generated by means of two non-isolated *DC-DC* converters and a LDO as shown in figure 5.23. The converters used are the NNL05-9C [98] from *C&D Technologies*. The input voltage range is from 4 V to 5.5 V and it has an adjustable voltage output (between 0.75 V and 3.3 V) with a maximum current of 5 A. The voltage output has been adjusted to 1.2 V in one converter (by means of *R4* and *R5*) and to 3.3 V in the other one (by means of *R9* and *R10*). The value of the resistance for adjusting the output voltage to a specified level is given in [98]. The load voltage regulation over a 0-100 % of load variation is 1% of the output voltage while the line voltage

regulation is 1% of the output voltage. Its switching frequency is 300 kHz with an output ripple and noise over a 20 MHz bandwidth is below 30 mV peak-to-peak. For stability purposes, a 330 μF tantalum (low ESR) capacitor (*C1* and *C9*) has been placed at the input of each converter.

Two filters (at input and output) have been implemented for each NNL05-9C, as can be seen in figure 5.23. The input filter is a second order damped pi-filter. The cut-off frequency of the filter has been fixed to 27.7 kHz by choosing the appropriate values for *L* and *C* (*L1*, *C2*, *C3*, *L3*, *C10* and *C11* in figure 5.23 for the input filters), around one decade below the switching frequency of the converter (300 kHz). The output filter is a second order damped pi-filter. The cut-off frequency of the filters has been fixed to 27.7 kHz as well. Pi-filters have been chosen for the output since it is expected that the digital hardware biased with this supply level is going to generate a considerable noise level. Therefore, the aim is to reduce the output ripple and noise generated by the switching converter as well as to reject the generated noise by the digital hardware. The filter capacitors are ceramic capacitors and the *DC* blocking capacitors are tantalum capacitors. Inductances with low values have been selected (3.3 μH) with a rated current of 5.5 A (for an input full load expected current of 3700 mA), a *DC* resistance of 12 mΩ and a self-resonance frequency of 50 MHz. From the 3.3 V digital supply level, a 2.5 V digital supply level is generated by means of the LP38691 [99] LDO linear regulator from *National Semiconductor*. It is a LDO linear regulator with a typical dropout of 250 mV at 500 mA load current. The capacitors at the input and output of the regulator are placed for the regulator stability and for noise minimization, and they have the recommended values from the manufacturer.

As it has been explained along this chapter, bypassing and decoupling capacitors have been placed close to each IC in order to reduce the noise produced in the power supply line by the digital hardware and to protect the analogue circuits from the digital noise. Furthermore, as it will be detailed in the following section, separated power planes and grounds have been designed in order to minimize the noise by reducing the inductance which would find the supply current. Moreover, the decoupling capacitors have been placed as close as possible to the corresponding IC power pins.

As a general rule for the analogue circuits, a second order filter has been implemented for each analogue power pin of each analogue IC with a ferrite in series and two parallel capacitors. One is a ceramic capacitor of low capacitance for the higher frequencies and the other one is a tantalum capacitor of higher capacitance for the lower frequencies. For the digital circuits, a variable number (depending on the IC) of ceramic capacitors with low capacitance value (from 1 nF

to 4.7 µF) in parallel will cope with the higher frequencies as decoupling capacitors at each IC power pin. Furthermore, a tantalum capacitor with higher capacitance value also in parallel will cope with lower frequencies as bypassing capacitor at each IC power pin. The placement of these capacitors with regard to the corresponding IC power pin must be considered specially in the digital case. As a general rule, the decoupling capacitors should be as close as possible to the corresponding IC power pin while the bypassing capacitor can be further away from the corresponding IC power pin. By reducing the distance between the capacitor and the device the path for the demanded current spikes will be reduced, thus reducing the inductance of that path and, therefore, the noise circulating across the power supply planes. This is more critical for the decoupling capacitors because they cope with the higher frequencies compared to the bypassing capacitors.

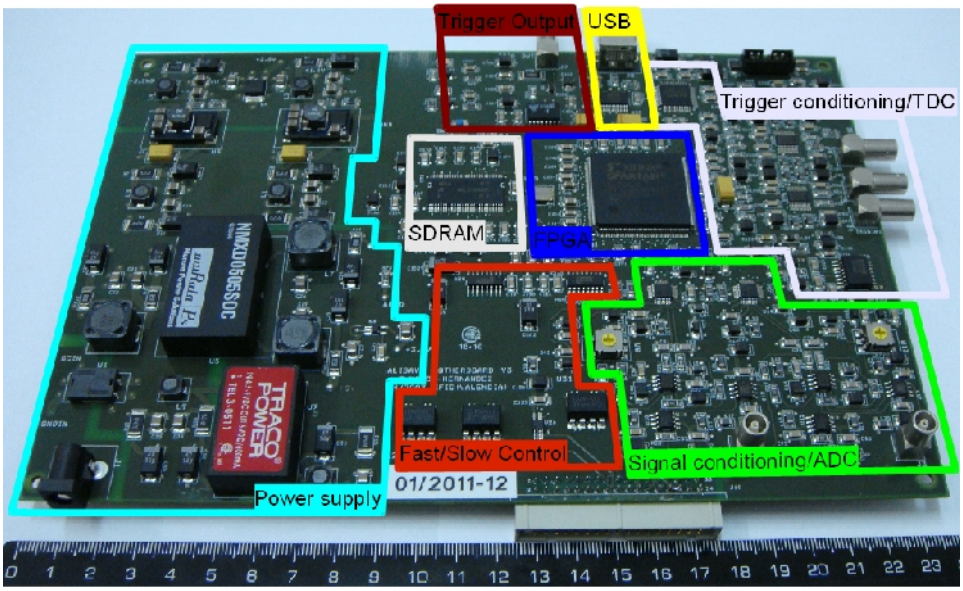

Figure 5.24. *Picture of the mother board PCB with the main hardware blocks marked.*

5.2.10. PCB and connection system

All the hardware described in the last sections has been physically implemented in a PCB as it is shown in the figure 5.24. The mother board PCB has a size of 235 x 160 mm. The complexity of the design due to multiple power voltages and ground planes, signal integrity and noise considerations as well as the density of traces made necessary the use of a multilayer PCB. A six layer PCB structure has been designed with the following layer stack-up from top to bottom,

- layer 1 (top): component side. Separated analogue and digital signals, some of them with controlled impedance. Separated signals for the power supply bock.
- Layer 2 (inner): split analogue, digital and daughter board ground planes. Separated ground planes for the power supply block.
- Layer 3 (inner): split analogue (±5 V and 3.3 V), digital power planes (1.2 V and 2.5 V) and daughter board power (5 V) planes. Separated power planes for the power supply block.
- Layer 4 (inner): digital power plane (3.3V) and daughter board power (5 V) planes. Separated power planes for the power supply block.
- Layer 5 (inner): digital ground plane and daughter board ground plane. Separated ground planes for the power supply block.
- Layer 6 (bottom): solder side. Digital signals, some of them controlled impedance signals as LVPECL and LVDS. Separated signals for the power supply bock.

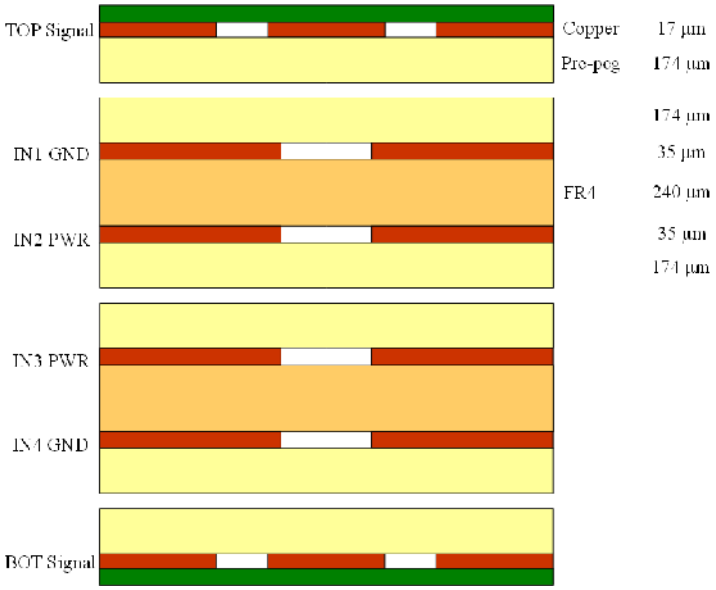

Figure 5.25. *Cross section of the mother board PCB showing its layer structure and distribution.*

The layer structure and distribution is shown in the figure 5.25. The PCB structure is composed of two cores with a FR4 (fibreglass and epoxy) sheet and two external copper layers. These two cores are combined with two outer copper layers and pre-peg (also fibreglass and epoxy) sheets in order to crate the PCB

structure. The unification is carried out through high temperature pressing process. The PCB layout uses four layers exclusively for ground and power planes, being the planes only interrupted by vias that connect the different layers. This structure reduces the impedance of the power to the ground in the whole PCB by allowing the shortest return path for the signals. Therefore the inductance is reduced, improving the response to high frequency signals and reducing noise levels. The power supply system uses the same physical PCB although is isolated from the rest of the hardware. Thus, the six layers are used for the supply system signals, maintaining the inner layers for power and ground planes and the outer layers for signals.

Before a PCB is laid out, care must be taken to place components properly on the PCB in order to minimize the EMI problems. Therefore, low level analogue, high-speed digital, and noisy circuits (*i.e.* circuits with high switched current level, as power supply *dc-dc* converters) must be separated to limit coupling between the subsystems to a minimum [47-49]. When placing components, close attention must be paid to the potential routing of circuits between subsystems.

In the design of the mother board PCB, these guidelines have been followed and the active components where firstly classified as belonging to power supply circuits, digital, analogue or mixed-signal (ADCs and DAC). Then, they were carefully placed following the previous separation rule, as it shown in the figure 5.26. The mixed-signal components have analogue signals and digital signals and they have been placed in the border between the digital area and the analogue area so that the analogue signals will be in the analogue area and the digital ones in the digital area. Furthermore, it is very important that signals belonging to different areas do not cross to other areas.

The ground layout is especially critical. In fact, the ground can be considered the foundation of all good PCB designs. Most EMI problems can be resolved by using practical and efficient grounding methods [47-49]. Understanding the mechanisms that generate ground noise is critical to minimizing ground interference. All ground paths have some finite impedance. As with all circuits, current flow must return to its source. Current flowing through finite impedance in the ground lines will cause a voltage drop. These voltage drops are the cause of interference in the ground system. As system frequencies increase, the resultant interference in the ground system also increases. Elementary circuit theory says that a change in a conductor current multiplied by the inductance of the conductor produces a voltage. High-frequency digital systems create current spikes when transistors are switched on and off. Analogue systems create current spikes when load currents change. Faster systems produce faster rise times increasing the noise magnitude. Most digital

systems have a higher immunity to noise than analogue. Low levels of noise in the ground system can severely affect the performance of low-level analogue circuits.

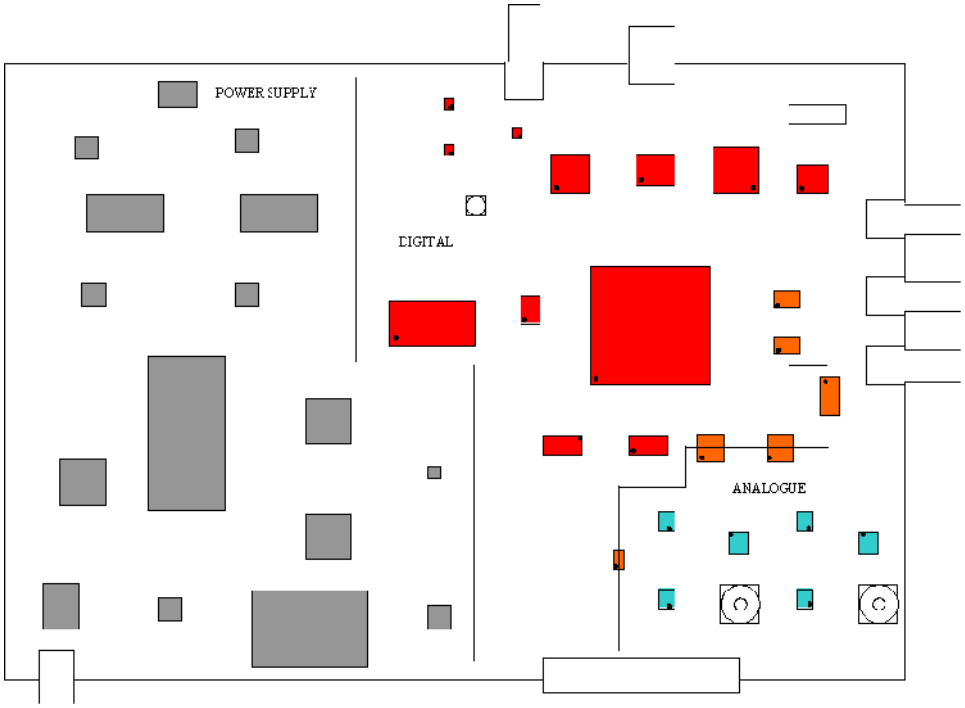

Figure 5.26. *Components placement at the mother board PCB. The power supply components (in grey), the digital components (in red) and the analogue components (in blue) have been separated in different zones. The components which have analogue and digital interface (in orange) like ADCs and DACs, have been situated in the border between the digital area and the analogue area.*

Noise can be coupled into other circuits by common impedance [47-49]. Figure 5.27 illustrates the coupling problem. The voltage at the summing point of the two signals is due to currents and inductances found in the analogue and digital signals. The noise created is now shared due to common impedance, *Z3*, between the two signals. A *dc* offset is created between the system ground point and the summing point. In digital systems, this offset is dynamic and produces a high-frequency *ac* component of noise which will affect low-level analogue circuitry.

One advantage of a well-thought-out ground system is providing protection against unwanted interference without additional board cost except for engineering design time. The basic objective of a good ground system is to minimize noise voltage from currents flowing through ground impedances. Signal grounds that

have low-impedance paths to return to the source should be created. This can be accomplished by determining the type of circuitry used and the operating frequency of the system. As mentioned earlier, circuits of different nature should be separated and their ground returns should not be mixed together. Similar circuits should be placed together.

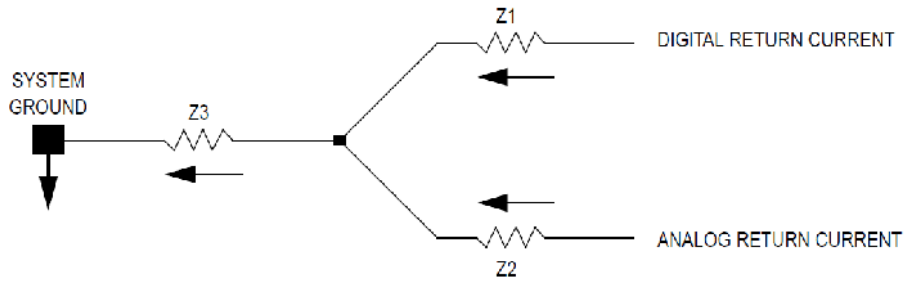

Figure 5.27. *Sketch showing the common impedance coupling.*

High-speed digital circuits must provide low-impedance pathways for all return signals. The ground system must be designed to include as many parallel pathways to ground as possible. This will decrease the inductance of the ground return. If this concept is taken to the limit, a ground plane will be created. Therefore, ground planes are optimal for this purpose. Moreover, single-point grounding should be used among grounds of different nature circuits to protect susceptible circuits (as analogue ones) against noisy circuits (as digital and power supply). Single-point or star-point grounding ties all ground traces to the terminal ground point. Finally, it must be remembered that current will flow back to its source eventually. In some cases, the return path will create a large loop that is highly susceptible to electromagnetic radiation and will couple noise into the ground system. As a general rule, the size of all ground loops must be decreased as much as possible which it is carried out by using ground planes.

In this system, following these guidelines, two layers have been dedicated to system ground planes. The first inner layer from the top (layer 2) has a digital ground plane, an analogue ground plane and a daughter board supply ground plane as well as ground planes for the power supply block. All these planes have been separated in order to provide different paths for different ground currents, as it can be seen in figure 5.28. The separation of the different ground planes corresponds to the different areas delimited in the component placement process. The mixed-signal components, which have digital and analogue grounds, have been placed just in the separation between the digital ground plane and the analogue ground plane. The fourth inner layer of the mother board PCB (layer 5) has been also dedicated to

ground planes, a digital ground plane, a daughter board supply ground plane as well as ground planes for the power supply block (figure 5.29). The same ground planes of different layers are connected with vias. Finally, the digital and analogue ground planes have been connected close to the power supply for a star-point configuration by means of a 0 Ω resistor. The analogue ground and the daughter board supply ground have been also connected at the IDC connector for a star-point configuration.

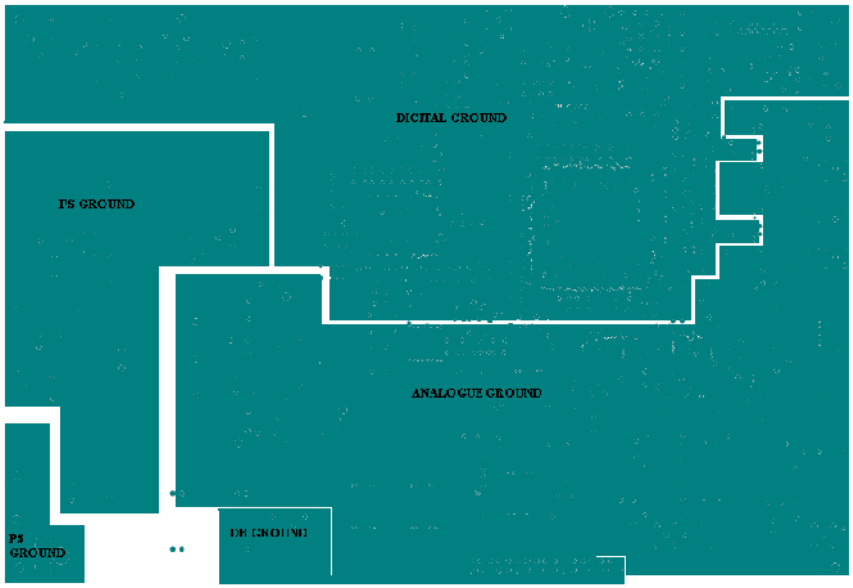

Figure 5.28. *Second layer of the mother board PCB where it is shown the ground planes distribution.*

The outermost inner layers have been used for ground planes since signal traces have been placed in the external layers. Therefore, the analogue signals have been placed in the top layer while the digital signals have been placed both in the top and the bottom layer in order to control the trace impedance when required as well as to minimize the loops formed by the signals and its return ground path. Furthermore, the noise problems are reduced considerably with this configuration, since each signal will have its return ground path just below along its corresponding ground plane and each group of signals (digital, analogue and power supply) have been placed in its corresponding area (without any crossing between different areas).

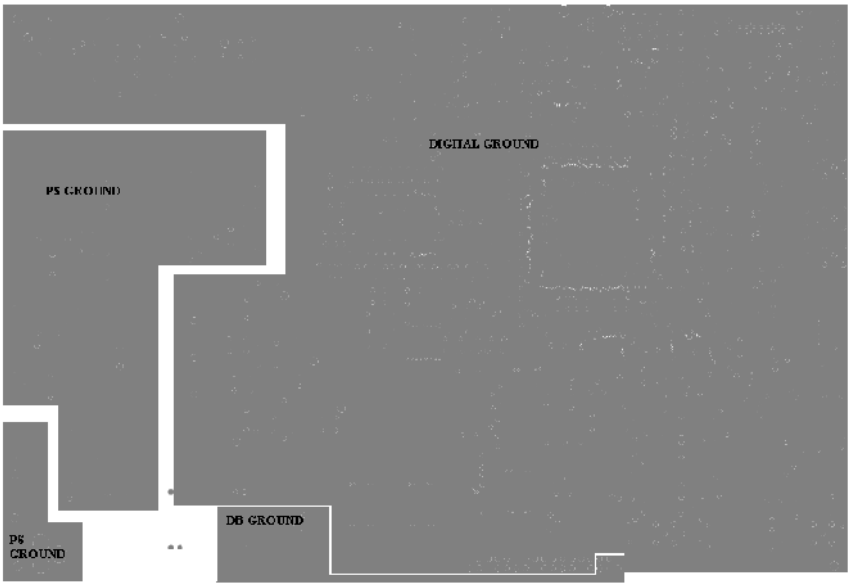

Figure 5.29. *Fifth layer of the mother board PCB where it is shown the ground planes distribution.*

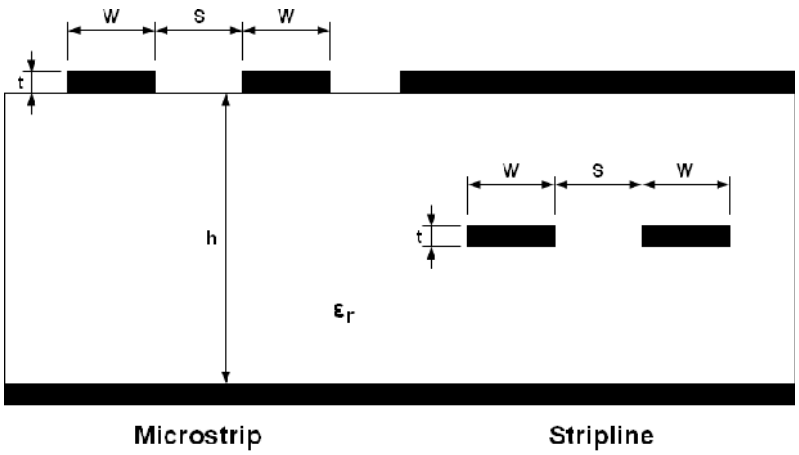

Figure 5.30. *Microstrip and stripline structures for controlling impedance of PCB traces.*

On the other hand, different signal traces have controlled impedance for maintaining signal integrity by matching impedances as it has been explained during this chapter. These signals have been treated as microstrip lines because of the PCB layer structure. The microstrip and the stripline structure of PCB traces is shown in figure 5.30. The medium is characterized by a dielectric constant ε_r, S

denotes the distance between traces, h denotes the thickness of the board, W denotes the trace width and t denotes the trace thickness. Two equations for the microstrip structure [100] have been applied in order to know an approximation of the trace impedance, Z_0, or the differential impedance, Z_{DIFF}, if two traces are considered,

$$Z_0 = \frac{60}{\sqrt{0.457 \cdot \epsilon_r + 0.67}} \cdot \ln\left[\frac{4 \cdot h}{0.67 \cdot (0.8 \cdot W + t)}\right] \quad (5.17)$$

$$Z_{DIFF} = 2 \cdot Z_0 \cdot (1 - 0.48 \cdot e^{\frac{-0.96 \cdot S}{h}}) \quad (5.18)$$

where Z_0 and Z_{DIFF} are in ohms and the rest of dimensions in cm. Different signals in the mother board have traces with controlled impedance. Digital fast signals like the clock signals, the LVDS differential signals and the LVPECL differential signals have 100 Ω. The trigger output digital signal has 50 Ω. The differential analogue input signals have 100 Ω while the trigger input analogue signals and the analogue output signals for the oscilloscope have 50 Ω.

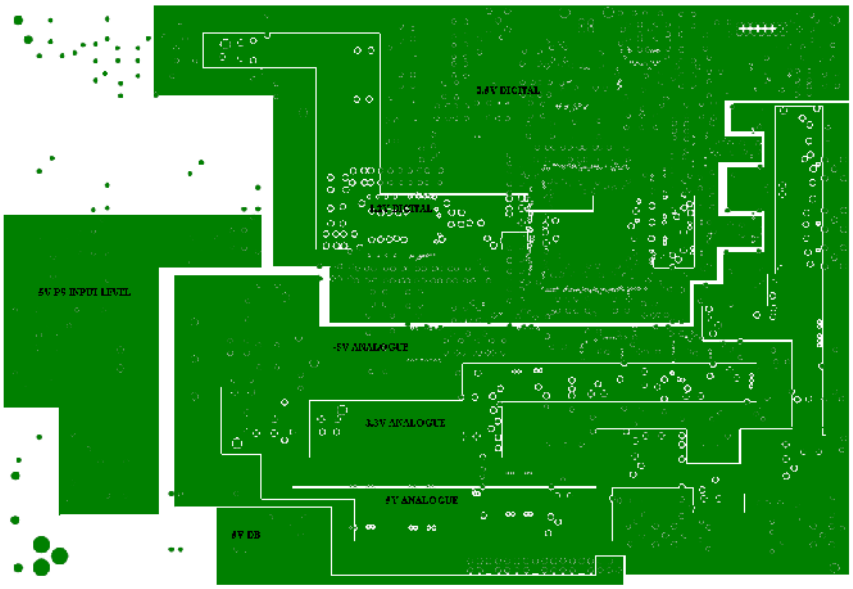

Figure 5.31. *Third layer of the mother board PCB where it is shown the power planes distribution.*

For the supply levels distribution, the same rules as for the ground distribution

have been considered. Therefore, two inner layers have been used for placing power supply planes. The supply planes must be as close as possible to the corresponding ground plane. Moreover, the power supply plane and the corresponding ground plane should match considering their areas. Therefore, the second inner layer (layer 3) has the analogue power planes (±5V and 3.3V), the digital power planes (1.2V and 2.5V), the daughter board power supply level (5V) as well as the power planes for the power supply block as it is shown in figure 5.31. The third inner layer (layer 4) has a digital power plane (3.3V), the daughter board power supply level (5V) and the power planes for the power supply block as it is shown in figure 5.32.

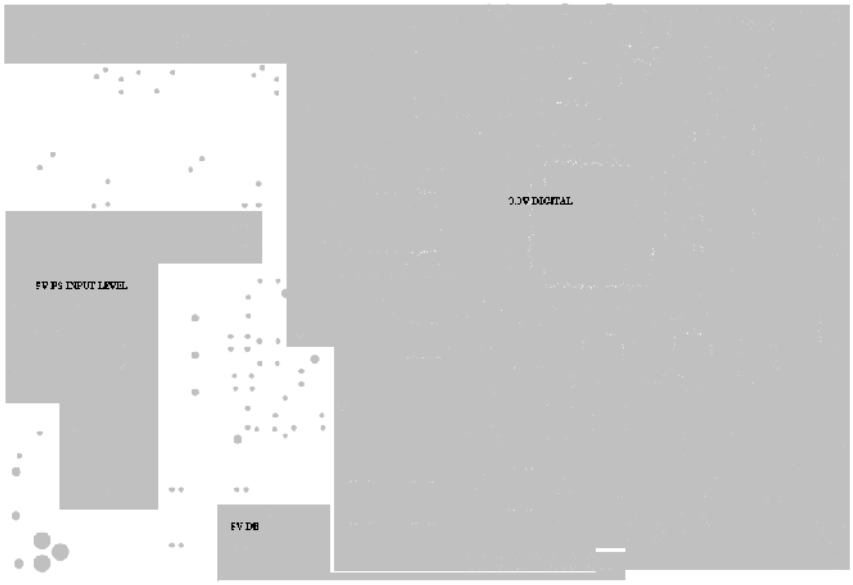

Figure 5.32. *Fourth layer of the mother board PCB where it is shown the power planes distribution.*

The decoupling and bypassing capacitors of the system have been connected to the corresponding power and ground planes using vias. The parasitic inductance of current paths in the PCB has two distinct sources: the capacitor mounting, and the power and ground planes of the PCB. In this context, the mounting refers to the capacitor's solder land on the PCB, the trace between the land and via, and the via itself. The vias, traces, and pads of a capacitor mounting contribute anywhere from 300 pH to 4 nH of inductance depending on the specific geometry [90]. Since the inductance of a current path is proportional to the area of the loop the current traverses, it is important to minimize the size of this loop. The loop consists of the path through one power plane, up through one via, through the connecting trace to

the land, through the capacitor, through the other land and connecting trace, down through the other via, and into the other plane, as shown in figure 5.33 (left side). By shortening the connecting traces, the area of this loop is minimized and the inductance is reduced. Similarly, by reducing the via length through which the current flows, loop area is minimized and inductance is reduced. Therefore, for the decoupling and bypassing capacitors (figure 5.33 right side) the number of vias necessary has been doubled and the traces shortened (also wider traces have been used) in order to reduce the parasitic inductance of the structure, thus reducing the noise impact.

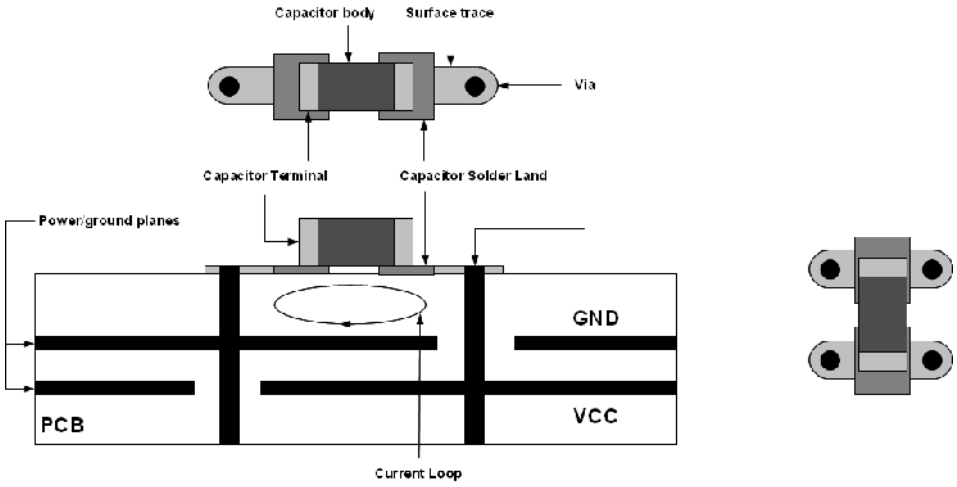

Figure 5.33. *Cutaway view of PCB with capacitor mounting (left side) and optimal capacitor mount geometry (right side).*

Regarding the connection system of the mother board, several connectors have been used in the PCB.

- Connection with daughter board: 34 way IDC connector.
  - Daughter board power levels: *+5V* and *GND*.
  - *Fast contro*l signals (LVDS): *Reset*, *TestPulse*, *Clk*, *Trigger*, *DataValid0* and *DataValid1*.
  - *Slow control* (I^2C): *SDA*, *SCL* and *DGND*.
  - Thermistor signals: *NTC* and *AGND*.
  - Analogue differential signals from Beetle chips: *AO0-nAO0* and *AO1-nAO1*.
- Scope connection: LEMO 00 EPA.00.250.NTN (straight). Two output analogue signals.

- Trigger inputs/output: LEMO 00 EPL.00.250.NTN (right angle).
  - TRIG OUT.
  - TRIG IN1, TRIG IN2 and TRIG PULSE IN.
- USB: USB Receptacle- Right Angle Type B.
- Power supply (*dc*): 2.1 mm PCB *dc* Power Socket.
- FPGA configuration (JTAG): 14 way straight pin shrouded header.

5.3. FPGA logic

The main function of the FPGA is implementing all the logic in order to control the hardware and to communicate with the host computer. The block diagram implemented in the FPGA can be seen in figure 5.34. The *CFSM* (*central finite state machine*) block controls the hardware by interpreting the orders that the software sends by USB. Depending on the current state the *CFSM* block will use different blocks and it will enable the communication among those blocks.

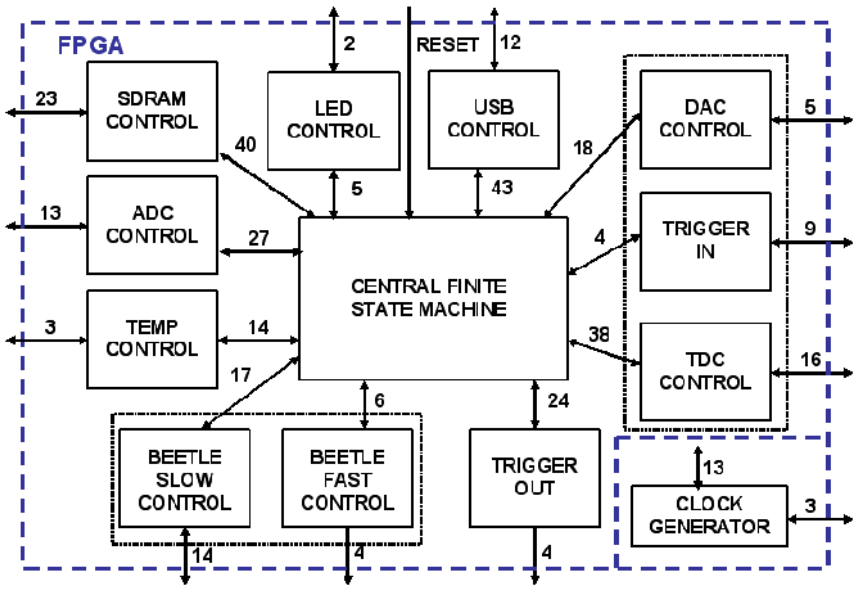

Figure 5.34. *Generic block diagram of the logic implemented in the system FPGA.*

The *Beetle Slow Control* block controls the I²C bus in order to program the Beetle configuration registers. The *Beetle Fast Control* block generates the LVDS output signals (*Clk*, *Reset*, *Trigger*, *TestPulse*) for the Beetle operation. The *TestPulse* is generated from an internal calibration signal. The *Trigger* signal is generated from an internal TRIG_L signal (in case of laser setup) or from an

internal TRIG_R signal (in case of radioactive source setup) taking into account both the Beetle analogue pipeline latency and the particular synchronization delay.

The *Trigger Out* block is used for the laser setup and it produces an external trigger signal (TRIG OUT) and an internal trigger signal (TRIG_L) for the *Fast Control* block. This block also controls the programmable delay circuit. In this way, by programming a variable delay (up to 255 ns in 1 ns steps) the system can acquire a specific point of the Beetle front-end analogue pulse.

The *Trigger In* block generates an external and internal trigger signal (TRIG and TRIG IN, respectively) from signals SIN1 and/or SIN2 (discriminated signals from two photomultipliers), PPOS (external positive pulse) or PNEG (external negative pulse) coming from the discriminators. The coincidence of SIN1 and SIN2, as well as which inputs will be used, can be programmed. The *DAC Control* block programs the DAC that supplies the four voltage thresholds for the trigger conditioning comparators. The *TDC Control* block interfaces the TDC, which measures the time from the TRIG leading edge to a STOP leading edge signal (derived from a 100 ns periodic signal in case of TRIG leading edge). It also generates a TRIG_R signal (for the *Fast Control* block) related in time to the STOP signal when the last will be active.

The *ADC Control* block interfaces the ADCs, it reads the digitalized data frames when the corresponding *DataValid* signal will be active and it stores these frames in an internal FIFO RAM. The *SDRAM Control* block communicates with the SDRAM. It also implements the read/write data and control interface with the *CFSM* block. The *USB Control* block implements the communication between the USB chip and the *CFSM* block. The *Clock Generator* block supplies the FPGA internal clock signals and reset signals. The *Temperature Control* block reads the digital conversion of the thermistor signal from the digital converter by serial interface. The *LED Control* block activates a red LED or a green LED depending on its input code value. This is used to show to the user the state of the system.

The *CFSM* and the *SDRAM Control* blocks have been implemented using an embedded processor. A *Microblaze* processor has been used [101]. The interface used with the embedded processor is composed by six FIFO memories. Four of them have been implemented for direct communication with the *ADC Control* block and *USB Control* block and the other two have been implemented for general communication with the rest of the blocks. For this communication an *Arbiter* block has been designed. This block implements the direct data communication with the *USB Control* block and the *ADC Control* block. It also implements different registers for capturing the data signals coming from the rest of the blocks

or for generating the signals required to control all the blocks.

5.3.1. *ADC Control* block

There is one *ADC Control* block in the FPGA for each ADC present in the mother board. These two controllers have three purposes. First, each *ADC Control* block interfaces the corresponding ADC of the mother board by generating both the PD (power down) signal and the SCLK signal (40 MHz sampling clock, see section 5.2.1). Second, each *ADC Control* block reads the digitized data frames during an event, from the D9-D0 bus, and stores the data in an internal FIFO memory. Therefore, the block also controls the FIFO memory interface. Finally, each *ADC Contro*l block synchronizes the generation of the SCLK output signal from the corresponding *DataValid* input signal leading edge, which will inform that a single readout from the corresponding Beetle chip will be present at the ADC inputs in 25 ns and it must be sampled and digitized.

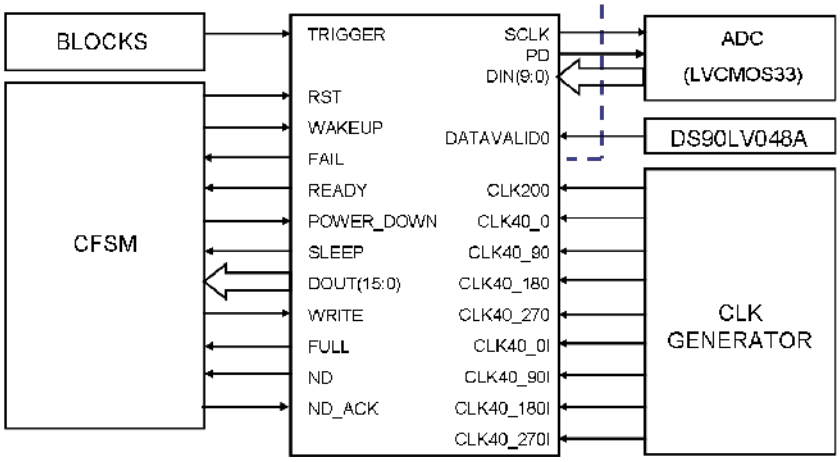

Figure 5.35. *ADC Control block and its connections with external circuits and other internal FPGA blocks.*

The *ADC Control* block is shown in figure 5.35, where the different signals of the block are detailed. There are several external signals for interfacing the corresponding ADC and for receiving a fast control signal from the corresponding Beetle chip,

- SCLK: sampling clock signal. The frequency is fixed to 40 MHz. The samples are taken at the falling edge.
- DIN(9:0): this bus is the ADC output digitized data which are read and

stored in the internal FIFO memory. The data are sampled with the falling edge of SCLK and they are valid on the leading edge of the sampling clock. There is an internal latency of 5.5 clock cycles from the data are sampled until the data are valid at the output.

- PD: power down signal. When high the ADC will be in power down mode. When low the ADC is ready for normal operation.
- DATAVALID: this is a fast control signal (LVDS) generated by each Beetle chip. It will be high level 25 ns before the first header bit of the corresponding Beetle analogue signal will be generated. Therefore, this signal is used for synchronizing the ADC sampling clock with the Beetle analogue signal in order to sample the middle point of each multiplexed channel.

The rest of the signals of the block shown in the figure 5.35 are internal signals of the FPGA. There are different clock signals which are used for generating the SCLK clock signal as well as for clocking the sequential logic of the block,

- CLK40_0 (CLK_40_0I): reference clock signal of 40 MHz used by all the FPGA blocks. The signal CLK_40_0I is the same signal but it does not use clock resources (buffers and special clock paths) since it is connected to an external port.
- CLK40_90 (CLK_40_90I): clock signal of 40 MHz delayed 90 degrees or 6,25 ns regarding to CLK_40_0. The signal CLK_40_90I is the same signal but it does not use clock resources since it is connected to an external port.
- CLK40_180 (CLK_40_180I): clock signal of 40 MHz delayed 180 degrees or 12,5 ns regarding to CLK_40_0. The signal CLK_40_180I is the same signal but it does not use clock resources since it is connected to an external port.
- CLK40_270 (CLK_40_270I): clock signal of 40 MHz delayed 270 degrees or 18,75 ns regarding to CLK_40_0. The signal CLK_40_270I is the same signal but it does not use clock resources since it is connected to an external port.
- CLK200: clock signal of 200 MHz in phase with CLK40_0.

The rest of the signals interface the block with the *CFSM*. A TRIGGER signal is received from the *Fast Control* block when a trigger has been sent to the Beetle chips.

- RST: active high asynchronous reset signal for the block.
- WAKEUP: active high input signal for activating the block and the

corresponding ADC from a previous power-down.

- FAIL: active high registered output signal. It will be active if the last data has not been acquired fully.
- READY: active high registered output signal. It will be active when the block is ready for a new acquisition of data.
- POWER_DOWN: active high input signal for powering-down the corresponding ADC.
- SLEEP: active high registered output signal. It will be active when the block is in power-down state.
- DOUT(15:0): output data bus which will contain the acquired data from the ADC plus six filling bits (all bits from bit 15 to bit 10 are zeros). This bus is directly connected to the corresponding internal FIFO memory.
- WRITE: active high registered output signal. It will be active when the new data is present at the DOUT(15:0) bus in order to be stored in the corresponding FIFO.
- FULL: active high input signal for indicating if the FIFO is full.
- ND: active high registered output signal. It will be active when the new data has been acquired and stored in the corresponding FIFO.
- ND_ACK: active high input signal for acknowledging a new data output signal. It must be activated for deactivating a previous active ND signal.

The main logic block of the *ADC Control* is the SCLK generator. This block is able to generate the required sampling clock (SCLK) synchronized to the corresponding *DataValid* signal The synchronization is carried out automatically from the leading edge of the corresponding *DataValid* signal. The SCLK only is active for sampling the 128 mutiplexed Beetle analogue channels. This SCLK generator uses the clocks described above for this synchronization. The SCLK for each ADC is generated automatically depending on the delay between the *DataValid* leading edge and the reference clock (CLK40_0) in order to sample the middle-point of each multiplexed channel of the Beetle analogue signal. A 200 MHz clocked flip-flops are also used to additionally delay the sampling clock if required.

The *ADC Control* block has to synchronize the input data to the internal reference clock (CLK40_0) in order to store correctly the data into the internal FIFO, since the WRITE signal is a registered signal. The internal FIFO for each ADC block has a size of 128 by 16 bits in order to store a full readout each time. There are also two counters for timing and synchronization purposes. All this logic is controlled internally with a *finite state machine* (FSM) which also interfaces with the CFSM. The TRIGGER signal is used as redundant signal for error checking. Therefore, if the time passed from the leading edge of the TRIGGER signal and the

leading edge of the corresponding *DataValid* signal is greater than a safety threshold (10 μs), the ADC block would activate the FAIL signal.

5.3.2. *Fast Control* block

The *fast control* signals generated by this block are *Clk*, *Reset*, *Trigger* and *TestPulse*. These signals are generated in 3.3 V LVCMOS format and then converted to LVDS format by a hardware driver as explained in section 5.2.2. The *Fast Control* block is shown in the figure 5.36.

The external signals of the block (connected to FPGA output ports) are the *fast control* signals.

- BCLK: Beetle clock signal with a frequency of 40 MHz.
- RESET: Beetle chip active high asynchronous reset signal. When this signal is active the Beetle chip is completely reset (see chapter four for more details).
- TRIGGER: Beetle chip active high synchronous trigger signal. When this signal is active, at least during a clock cycle, a specific sampled data stored in the analogue pipeline of the Beetle chip is selected, multiplexed and read out through the analogue output of the chip.
- TESTPULSE: Beetle chip active high asynchronous signal. This signal is used to generate internal pulses in order to inject a specific charge into each input channel for calibrating the Beetle chip. This signal is used in conjunction with the TRIGGER signal for acquiring readouts for known injected charges.

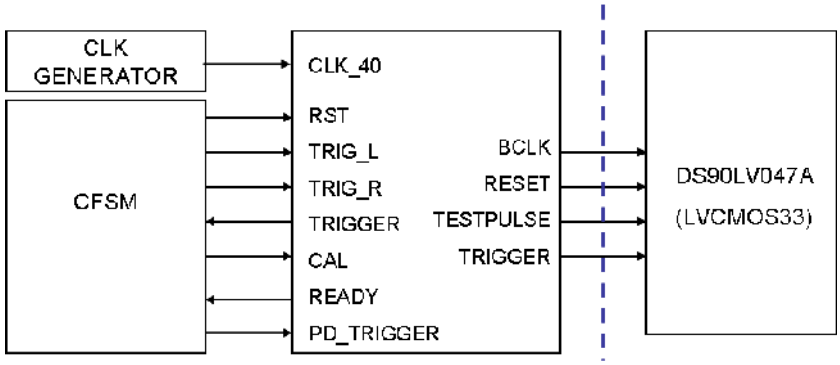

Figure 5.36. *Fast Control block and its connections with external circuits and other internal FPGA blocks.*

Although there are two Beetle chips on the daughter board, only one set of *fast control* signals is generated since these signals are shared by both Beetle chips. All the signals are registered signals in order to avoid glitches. The rest of the signals of the block shown in the figure 5.36 are internal signals of the FPGA which are connected to the *CFSM* and to the *Clock Generator*.

- CLK_40: reference clock signal of 40 MHz used by all the FPGA blocks.
- RST: active high asynchronous reset signal for the block.
- TRIG_L: active high synchronous pulsed signal. This input signal will be active during one clock cycle when a laser pulse has been triggered and the Beetle chips must be triggered to acquire the corresponding readout.
- TRIG_R: active high synchronous pulsed signal. This input signal will be active during one clock cycle when an input trigger from the radioactive source has been detected and the Beetle chips must be triggered to acquire the corresponding readout.
- PD_TRIGGER: active high synchronous pulsed signal. This input signal will be active during one clock cycle when the Beetle chips must be triggered to acquire a readout for calculating the pedestals, *i.e.* readout without any charge collected.
- CAL: active high synchronous pulsed signal. This input signal will be active during one clock cycle when the Beetle chips must be triggered to acquire a readout for calibration. In this case, both the *TestPulse* and the *Trigger* signals will be used in order to trigger and to acquire the corresponding calibration readout.
- TRIGGER: active high registered pulsed signal. This output signal will be active during one clock cycle when a *Trigger* signal has been generated to inform to the *CFSM*.
- READY: active high registered signal. This output signal will be high when the block is ready for accepting a new command from the *CFSM*.

The logic of this block has been implemented by means of a FSM which controls all the possible states of the system, reads the input signals and generates the output signals. Also a counter has been implemented in order to be used by the FSM. The *Trigger* signal will be generated, either in case of laser trigger, radioactive source trigger, calibration trigger or pedestals trigger, 128 clock cycles (Beetle chip latency) after a particular trigger has been activated. The *Trigger* signal is synchronous regarding the BCLK clock since it is sampled in the Beetle chip with the falling edge of this clock (see chapter four for more details). In the case of the laser setup, the TRIG_L signal will be delayed in the *Trigger Out* block according to the programmed delay for the synchronization.

5.3.3. *Slow Control* block

The *Slow Control* block is used to generate and to read back the *slow control* configuration commands for the Beetle chips. With these commands the Beetle chip are configured by writing in their configuration registers. These registers are read back in order to check that the Beetle chips have been configured correctly. The *Slow Control* block is shown in the figure 5.37. The *slow control* for the Beetle chip has been implemented by means of the I2C serial protocol (see chapter 4). The *slow control* bidirectional signals are generated with this block which acts as master transmitter and receiver for the I2C protocol,

- SDA: I2C serial data line open-drain bidirectional signal. With this signal data can be sent and received synchronously regarding the SCL line. A tri-state buffer has been used to implement this bidirectional signal (see figure 5.5).
- SCL: I2C serial clock line open-drain bidirectional signal. A tri-state buffer has been used to implement this bidirectional signal (see figure 5.5).

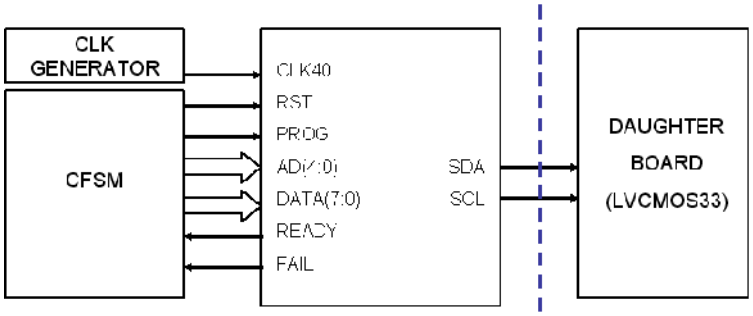

Figure 5.37. *Slow control block and its connections with external circuits and other internal FPGA blocks.*

Although there are two Beetle chips on the daughter board, only one set of *slow control* signals is generated since these signals are shared by both Beetle chips. All the signals are registered signal in order to avoid glitches. The rest of the signals of the block shown in the figure 5.37 are internal signals of the FPGA which are connected to the *CFSM* and to the *Clock Generator,*

- CLK40: reference clock signal of 40 MHz used by all the FPGA blocks.
- RST: active high asynchronous reset signal for the block.
- PROG: active high synchronous signal. This input signal will be active during one clock cycle if there are valid data in the AD(4:0) bus and DATA(7:0) bus to be programmed.

- AD(4:0): active high synchronous bus. This input bus is used to indicate to the block the address of a Beetle chip configuration register.
- DATA(7:0): active high synchronous bus. This input bus is used to indicate to the block the data to be programmed in a Beetle chip configuration register defined by the address bus.
- READY: active high registered signal. This output signal will be high when the block is ready for accepting new data from the CFSM.
- FAIL: active high registered signal. This output signal will be high when the block has detected that data has not been sent correctly either if a protocol error has been detected or if the data which has been read back does not agree with the data sent previously.

The core of this block is an I^2C communication core [102] which is operated using the so-called Wishbone interface [103]. Therefore, custom logic has been implemented to interface this core. This logic has been designed as a FSM which controls the core for configuration and operation (read/write) as well as implements the interface with the *CFSM*.

5.3.4. *Trigger In* block

The coincidence and multiplexing logic for the *Trigger Conditioning* block has been implemented by means of combinational logic whereas the pulse generation has been implemented with sequential logic. This logic has been grouped in the *Trigger In* block in the FPGA. Two signals have been introduced for the coincidence and the multiplexing, NE and CNIM. The signal NE will be high when just SIN1 or SIN2 are going to be used, otherwise the signal NE will be low when the coincidence of SIN1 and SIN2 are going to be used. The signal CNIM will be used to multiplex the photomultiplier signals (SIN1 and SIN2) from the auxiliary signals (PPOS and PNEG). Hence, CNIM will be high when PPOS or PNEG are going to be used whereas it will be low when any combination of SIN1 and SIN2 are going to be used. A schematic of the coincidence and multiplexing logic as well as the pulse generation logic is shown in the figure 5.38.

For controlling the pulse generation logic, there are three input digital signals: SET, CLK and EN. The SET signal will be the trigger pulse signal from the coincidence and multiplexing logic. This SET signal will have a pulse width which will vary depending on the discrimination and conversion level stages. The CLK signal will be a clock signal. This clock signal will have a frequency of 40 MHz, which is the main clock frequency of the system. The EN signal will be a signal to enable (when high) or disable (when low) the pulse generator. The detailed design of the pulse generator is shown in the figure 5.39. The output signal of this pulse

generation block will have a pulse width from one to two clock cycles (25-50 ns) depending on the phase between the SET leading edge and the CLK leading edge.

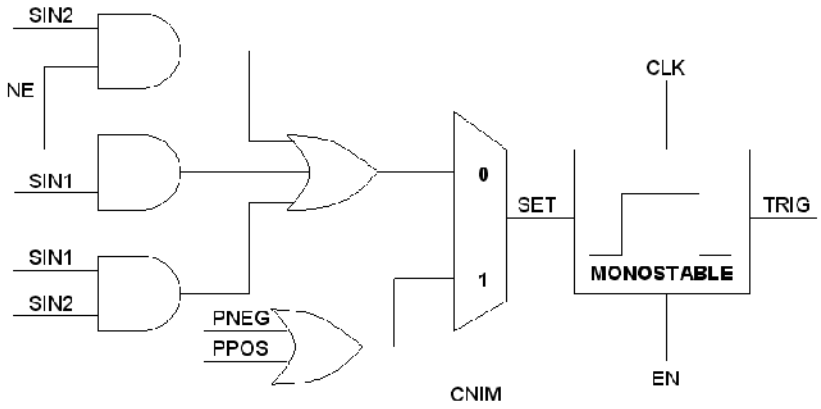

Figure 5.38. *Schematic circuit of the coincidence and multiplexing logic as well as the pulse generation logic implemented in the Trigger In block.*

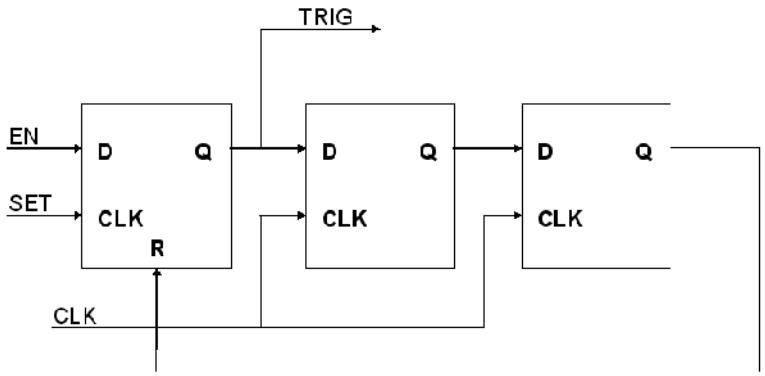

Figure 5.39. *Detailed schematic circuit of the pulse generation logic.*

Two options were considered for the implementation of this logic: a hardware implementation and a FPGA implementation. The hardware implementation would be carried out with discrete ECL logic gates and registers. The hardware implementation involves a greater cost and grater power consumption. The delay with this implementation will be lower but the dispersion delay over the whole operating range of temperature and power supply will be higher. Furthermore, the PCB complexity with this implementation will increase because more components should be used and the traces among the components should be impedance controlled. Therefore, all the logic was implemented in the FPGA. In table 5.5 a delay and delay dispersion comparison between the two proposed implementations

is shown. Finally, the Trigger In block with all its input and output signals is shown in the figure 5.40.

ECL				FPGA							
SIN1(SIN2) to TRIG		PPOS(PNEG) to TRIG		SIN1 to TRIG		SIN2 to TRIG		PPOS to TRIG		PNEG to TRIG	
td	3.1 ns	**td**	2.82 ns	**tdmax**	8.47 ns	**tdmax**	8.58 ns	**tdmax**	8.5 ns	**tdmax**	8.59 ns
				tdmin	6.95 ns	**tdmin**	6.94 ns	**tdmin**	6.98 ns	**tdmin**	7.06 ns
Δtd	2.96 ns	**Δtd**	2.76 ns	**Δtd**	1.52 ns	**Δtd**	1.54 ns	**Δtd**	1.52 ns	**Δtd**	1.53 ns

Table 5.5. *Delay and delay dispersion values (over the whole operating range of temperature and supply levels) for the discrete ECL implementation and the FPGA implementation.*

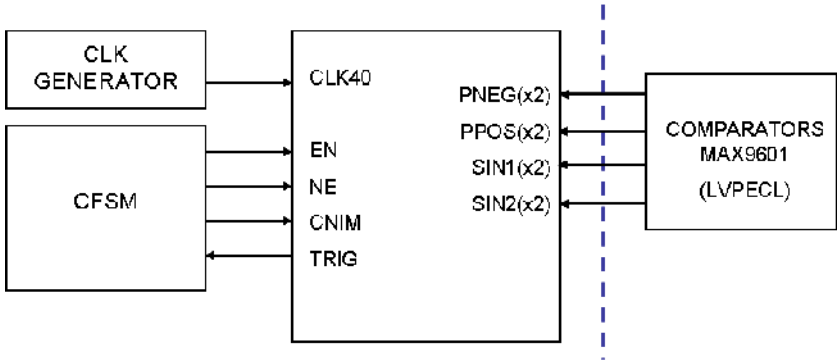

Figure 5.40. *Trigger In block and its connections with external circuits and other internal FPGA blocks.*

5.3.5. DAC Control block

The DAC control block has been designed in order to program the DAC circuit which generates the four thresholds required for the trigger inputs discrimination and level conversion carried out with the comparators of the *Trigger Conditioning* block. In the figure 5.41 the *DAC Control* block can be seen.

The external signals of the block are the SPI signals and other signals for programming and operating the DAC. These signals were explained in the section 5.2.3, where the operation of the DAC7614 was described. The rest of the signals

are internal signals connected to the *CFSM* and to the *Clock Generator* block:

- CLK40: reference clock signal of 40 MHz used by all the FPGA blocks.
- RST: active high asynchronous reset signal for the block.
- PROG: active high synchronous signal. This input signal will be active during one clock cycle if there are valid data in the AD(1:0) bus and DATA(11:0) bus to be programmed.
- DATA(11:0): active high synchronous bus. This input bus is used to indicate to the block the data to be programmed in a DAC register defined by the address bus.
- AD(1:0): active high synchronous bus. This input bus is used to indicate to the block the address of a DAC register for a specific threshold.
- READY: active high registered signal. This output signal will be high when the block is ready for accepting new data from the *CFSM*.

The logic of this block is basically a custom parallel to serial shift register controlled by means of an internal FSM which also implements the interface with the *CFSM* using the signals explained above.

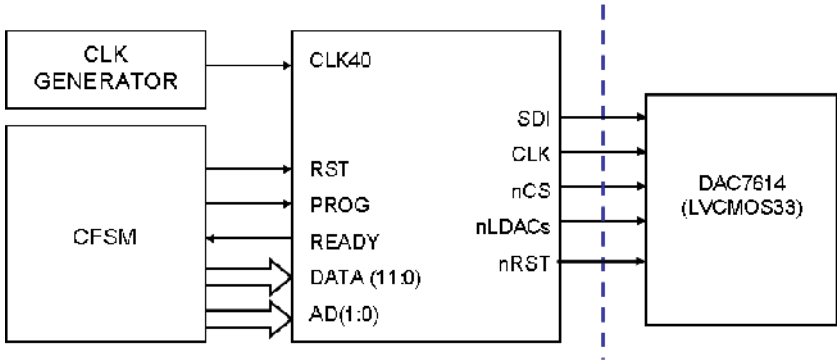

Figure 5.41. *DAC Control block and its connections with external circuits and other internal FPGA blocks.*

5.3.6. *TDC Control* block

The *TDC Control* block interfaces the external TDC, which is used with the radioactive source setup. The block configures the TDC for the required operation mode. It generates the START and STOP signals which drive the external TDC. The block also reads the measured data from this chip. The measured time will be the time passed between the leading edge of the START signal and the leading edge of the STOP signal. In the figure 5.42 the *TDC Control* block can be seen.

The external signals of the block are connected to the TDC of the mother board. These signals and the operation of the external TDC were explained in section 5.2.3. The START signal is connected directly to the TRIG_IN signal of the block. This signal is the processed signal from the different trigger inputs of the mother board, which are used for triggering the system in case of radioactive source setup. Therefore, this signal comes from the *Trigger In* block and it will be active when any of the trigger inputs of the mother board will be active, depending on the trigger input configuration scheme selected by the user. The STOP signal is generated by this block once a leading edge on the START signal has been detected previously. A STOP pulse of 25 ns will be generated from internal signal with a period of 100 ns for stopping the external TDC. Thus, it is possible to reconstruct the analogue pulse shape of the Beetle front-end chips from this time measurements, since the time measured between the leading edge of the START signal and the leading edge of the STOP signal will vary depending on the point of the analogue pulse shape of the Beetle chips which has been sampled and acquired.

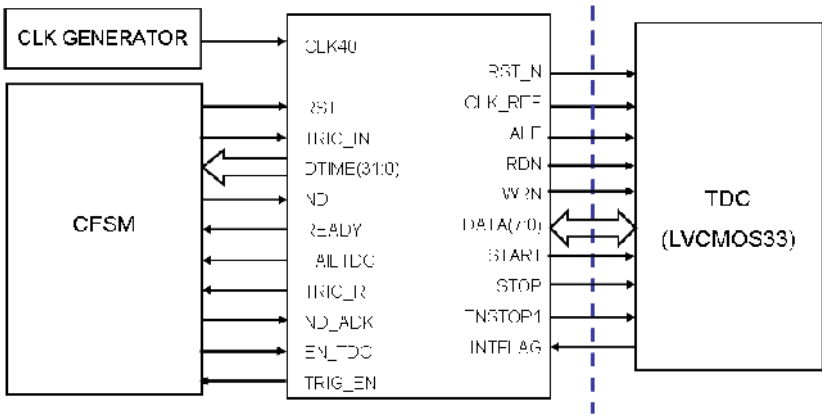

Figure 5.42. *TDC Control block and its connections with external circuits and other internal FPGA blocks.*

The rest of the signals are internal signals connected to the *CFSM* and to the *Clock Generator* block,

- CLK40: reference clock signal of 40 MHz used by all the FPGA blocks.
- RST: active high asynchronous reset signal for the block.
- TRIG_IN: active high asynchronous signal. This signal comes from the *Trigger In* block. It is a pulsed signal (pulse width from 25 to 50 ns) which will be active when any of the mother board trigger inputs will be active depending on the trigger input configuration scheme selected by the user.

- DTIME(31:0): active high registered bus. This output bus is used to output the last measured data from the TDC in order to be read by the *CFSM*.
- ND: active high registered signal. This output signal is used in conjunction with DTIME(31:0) to indicate to the *CFSM* that new valid data is ready to be read from the DTIME(31:0) bus.
- READY: active high registered signal. This output signal will be high when the block is ready for carrying out a new TDC measurement.
- FAILTDC: active high registered signal. This output signal will be high when the block has detected an error in the TDC operation during the last measurement or during configuration.
- TRIG_R: active high registered signal. This signal is connected to the *Fast Control* block. It will be active (25 ns pulse) when a STOP pulse has been generated, indicating to the *Fast Control* block that data must be acquired from the Beetle chips by activating the *Trigger fast control* signal taking into account the latency of the Beetle chips (128 clock cycles).
- ND_ACK: active high synchronous signal. This signal is used to indicate to the block that the last valid data at the DTIME(31:0) bus has been already read by the *CFSM*.
- EN_TDC: active high synchronous signal. This signal is used to enable the block for carrying out measurements with the TDC. If the signal is low, the TDC is configured and the block stays in a waiting state, waiting for a leading edge of this signal to start the operation.
- TRIG_EN: active high registered signal. This signal is used for enabling or disabling the pulse generator of the *Trigger In* block. It will be active if this pulse generator must be active.

The logic of this block has been implemented using a FSM which controls the operation of the block and performs the interfaces with the TDC as well as the *CFSM*. A pulse generator has been also designed in order to generate the STOP signal. This pulse generator is controlled by the FSM. There are also two counters for the FSM and the pulse generator.

5.3.7. *Trigger Out* block

The *Trigger Out* block is used with the laser setup. Its main function is generating the TRIG OUT signal which drives a laser source. This signal can be delayed externally by means of a digital delay circuit in 1 ns steps. Thus, the block also interfaces this digital delay circuit. The *Trigger Out* block is shown in the figure 5.43.

The external signals of this block are the interface signals for the digital delay circuit and the TRIG OUT signal, which were explained in detail in the section 5.2.4 as well as the operation of the digital delay circuit. The synchronization of the system with the laser setup is achieved both by delaying externally the TRIG OUT signal in 1 ns steps by means of the digital delay circuit and by varying the delay between the TRIG OUT signal and the TRIG_L signal in 25 ns steps. The TRIG_L signal is connected to the *Fast Control* block and it is used to activate the *Trigger fast control* signal after a fixed latency (128 clock cycles). Thus, it is possible to acquire a specific sampled point of the analogue pulse of the Beetle chip front-end. The 1ns delay is specified by means of FINE_COUNT(7:0) and the 25 ns delay is fixed by COARSE_COUNT(7:0) from a specific delay selected by the user.

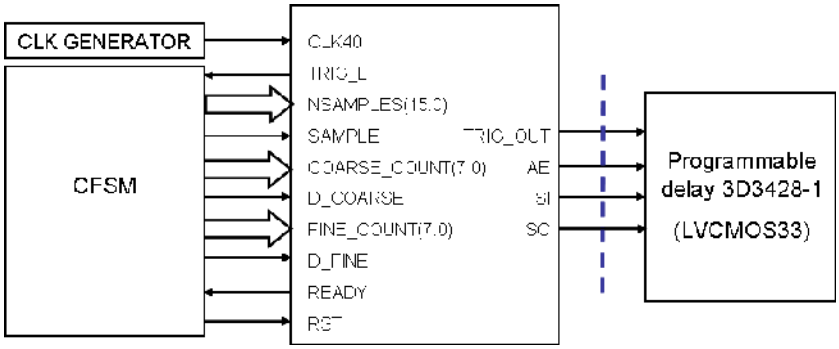

Figure 5.43. *Trigger Out block and its connections with external circuits and other internal FPGA blocks.*

The rest of the signals are internal signals connected to the *CFSM* and to the *Clock Generator* block,

- CLK40: reference clock signal of 40 MHz used by all the FPGA blocks.
- RST: active high asynchronous reset signal for the block.
- TRIG_L: active high registered signal. This signal is connected to the *Fast Control* block. It will be active (25 ns pulse) a number of 25 ns steps, specified by means of COARSE_COUNT(7:0), after a TRIG OUT pulse has been generated. It will indicate to the *Fast Control* block that data must be acquired from the Beetle chips by activating the *Trigger fast control* signal taking into account the latency of the Beetle chips (128 clock cycles).
- NSAMPLES(15:0): active high registered bus. This input bus is used to program the number of pulses at TRIG OUT that must be generated with a fixed frequency of 1 kHz.
- SAMPLE: active high synchronous signal. This input signal is used in

conjunction with NSAMPLES(15:0) to indicate to the block that new valid data is ready to be read from the NSAMPLES(15:0) bus.

- COARSE_COUNT(7:0): active high registered bus. This input bus is used to program the coarse delay (in 25 ns steps) between the leading edge of the TRIG OUT signal and the leading edge of the TRIG_L signal.
- D_COARSE: active high synchronous signal. This input signal is used in conjunction with COARSE_COUNT(7:0) to indicate to the block that new valid data are ready to be read from the COARSE_COUNT(7:0) bus.
- FINE_COUNT(7:0): active high registered bus. This input bus is used to program the fine delay (in 1 ns steps) at the external delay circuit. This fine delay will be used to delay the TRIG_OUT signal externally prior to drive the laser source.
- FINE: active high synchronous signal. This input signal is used in conjunction with FINE_COUNT(7:0) to indicate to the block that new valid data is ready to be read from the FINE_COUNT(7:0) bus.
- READY: active high registered signal. This output signal will be high when the block is ready to accept a new program data for programming a specific delay or a specific number of samples.

The logic of this block has been implemented using a FSM which controls its operation and performs the SPI interface with the external digital delay circuit as well as the interface with the *CFSM*. Two pulse generators have been also designed in order to generate the TRIG OUT and TRIG_L signals. These pulse generators are controlled by the internal FSM. There is also a parallel to serial shift register for the SPI interface controlled by the FSM. Three input registers have been implemented to store the data of the different input buses. Finally, three counters have been designed to be used by the different logic blocks.

5.3.8. *USB Control* block

The *USB Control* block interfaces the USB controller and implements the communication with the *CFSM*. From the point of view of this block both the internal interface and the external interface are two FIFO memories, one for writing data and another one for reading data. Therefore, this block controls the data flow between the internal *CFSM* and the external USB controller. The *USB Control* block can be seen in the figure 5.44.

The external signals of this block are all connected to the external USB controller. These signals and the operation of the USB controller were described in section 5.2.7. The rest of the signals are connected to the *CFSM* and to the *Clock*

Generator.

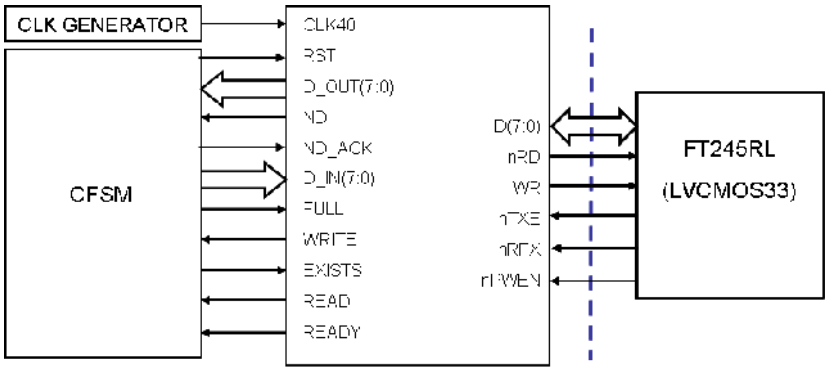

Figure 5.44. *USB Control block and its connections with external circuits and other internal FPGA blocks.*

- CLK40: reference clock signal of 40 MHz used by all the FPGA blocks.
- RST: active high asynchronous reset signal for the block.
- D_OUT(7:0): output data bus which will contain the read data from the USB controller. This bus is directly connected to the corresponding FIFO memory through the *Arbiter* block.
- WRITE: active high registered output signal. It will be active when the new data is present at the D_OUT(7:0) bus in order to be stored in the corresponding FIFO.
- FULL: active high input signal to indicate if the FIFO is full.
- D_IN(7:0): input data bus which will contain the data to write at the USB controller. This bus is directly connected to the corresponding FIFO memory through the *Arbiter* block.
- READ: active high registered output signal. It will be active when the new data is present at the D_IN(7:0) bus in order to be read from the corresponding FIFO.
- EXISTS: active high input signal for indicating that the FIFO has a new valid data for reading.
- READY: active high registered signal. This output signal will be high when the block is ready either to read data or to write data at the USB controller.
- ND: active high registered signal. This output signal is used to indicate to the *CFSM* that new valid data has been stored in the corresponding FIFO, *i.e.* the FIFO connected to D_OUT(7:0).
- ND_ACK: active high synchronous signal. This signal is used to indicate to the block that the last valid data at the FIFO connected to D_OUT(7:0)

bus has been already read by the *CFSM*.

The logic of this block has been designed using a FSM which implements the interface with the USB controller and the interface with the *CFSM*. The interface with the *CFSM* must control two internal FIFO memories with opposite flow data directions: one for the data that must be written to and the other one for the data that has been read from the USB controller.

5.3.9. *Temperature Control* block

A *Temperature Control* block has been designed to interface the external temperature converter for acquiring temperature readouts both with the radioactive source setup and with the laser setup. The *Temperature Control* block is shown in the figure 5.45. The external signals of this block are used to interface the temperature converter by means of SPI interface. The temperature converter acts as a slave while the block acts as a master. The operation of the temperature converter and their SPI signals were explained in the section 5.2.5 .

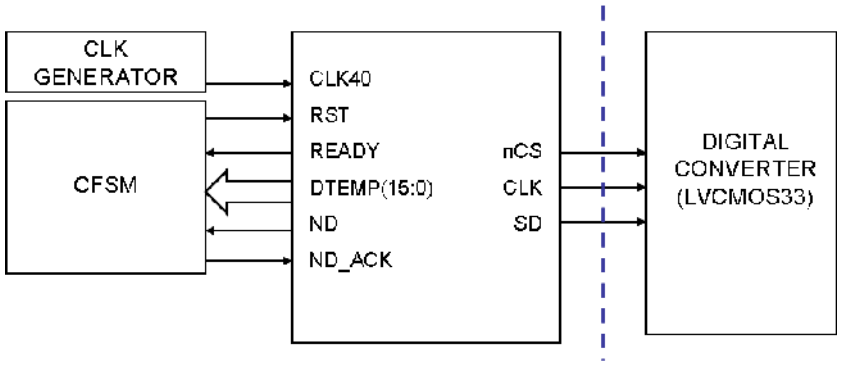

Figure 5.45. *Temperature Control block and its connections with external circuits and other internal FPGA blocks.*

The rest of the signals of this block are internal signals which are connected to the *CFSM* and the *Clock Generator,*

- CLK40: reference clock signal of 40 MHz used by all the FPGA blocks.
- RST: active high asynchronous reset signal for the block.
- DTEMP(15:0): active high registered data bus. Output data bus which will contain the read data from the external temperature converter, DTEMP(10:0), plus five filling bits (all zeros).
- READY: active high registered signal. This output signal will be high

when the block is ready to read data a new data from the temperature converter.

- ND: active high registered signal. This output signal is used to indicate to the *CFSM* that new valid data is available at the DTEMP(15:0).
- ND_ACK: active high synchronous signal. This signal is used to indicate to the block that the last valid data at the DTEMP(15:0) bus has been already read by the *CFSM*.

The logic of this block has been implemented using a FSM which controls the block operation and interfaces the *CFSM*. A serial-to-parallel shift register has been designed in order to read the data from the temperature converter by means of SPI interface. Also two counters have been implemented to be used by the FSM and the shift register. The temperature data is automatically refreshed at the DTEMP (15:0) bus taking into account the conversion rate of the temperature converter (0.47 Hz).

5.3.10. *LED Control* block

The LED Control block drives two external LEDs to inform to the user about the state of the system. It is a simple block which can be seen in the figure 5.46. The two external signals of the block drive two LEDs and they were explained in the section 5.2.8 of this chapter.

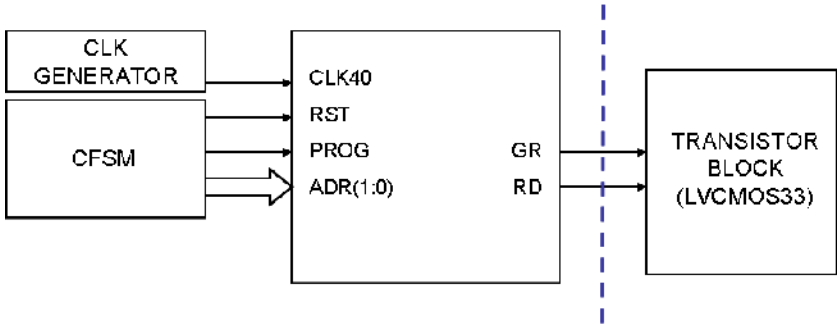

Figure 5.46. *LED Control block and its connections with external circuits and other internal FPGA blocks.*

The rest of the signals of the block are internal signals which are connected to the *CFSM* and the *Clock Generator,*

- CLK40: reference clock signal of 40 MHz used by all the FPGA blocks.
- RST: active high asynchronous reset signal for the block.
- ADR(1:0): active high synchronous data bus. Input data bus which will

contain the data to program which LED must be turned on, the red one ('01'), the green one ('10'), both ('11') or neither ('00').
- PROG: active high synchronous signal. This input signal is used to indicate to the block that there are new valid data at the ADR(1:0) to be read.

The logic of this block has been designed as a FSM which controls the external signals and the interface with the *CFSM*.

5.3.11. Embedded system

The *CFSM* and the *SDRAM Control* block of the figure 5.34 have been implemented using an embedded system in the FPGA. This embedded system consists of a software microprocessor core, a SDRAM controller peripheral connected to the microprocessor and a fast interface for the communication of the system with the rest of the custom logic blocks. The blocks of the embedded system have been designed and tested by the manufacturer of the FPGA and most of them are free. These blocks have to be tailored in order to be used in a specific application and connected in a specific way. All this can be done by means of a software package provided by the FPGA manufacturer as well.

Using this embedded system gives great flexibility since the logic design of the system can be kept while the functionality of the system can be changed by means of modifying the firmware of the processor. Therefore, the system functionality can be updated easier compared with a custom hardware solution, where the functionality must be changed by changing the logic design. Furthermore, the system is more reliable since the logic design of the embedded system blocks have been previously tested by the FPGA manufacturer and only the firmware is fully developed for each design.

The embedded microprocessor used in this system is the *Microblaze* software processor core [101]. A software processor core means that the usual FPGA hardware resources are used to build the processor hardware core whereas a hardware processor core is designed physically in the FPGA chip, so the FPGA in this case is more complex and expensive. The *Microblaze* is a 32-bit *reduced instruction set computer* (RISC) optimized for implementation in Xilinx FPGAs. The block diagram of this processor can be seen in the figure 5.47.

The processor core is highly configurable. The processor fixed feature set includes thirty-two 32-bit general purpose registers, 32-bit instruction word with three operands and two addressing modes, 32-bit address bus and single issue pipeline. *Microblaze* has a *Harvard* memory architecture, *i.e.* instruction and data

accesses are done in separate address spaces. Each address space has a 32 bit range. The instruction and data memory ranges can be made to overlap by mapping them both to the same physical memory. The processor does not separate between data accesses to I/O and memory (*i.e.* it uses memory mapped I/O). The processor has up to three interfaces for memory accesses: *Local Memory Bus* (LMB), *On-Chip Peripheral Bus* (OPB), and *Xilinx CacheLink* (XCL). The memory maps on these interfaces are mutually exclusive.

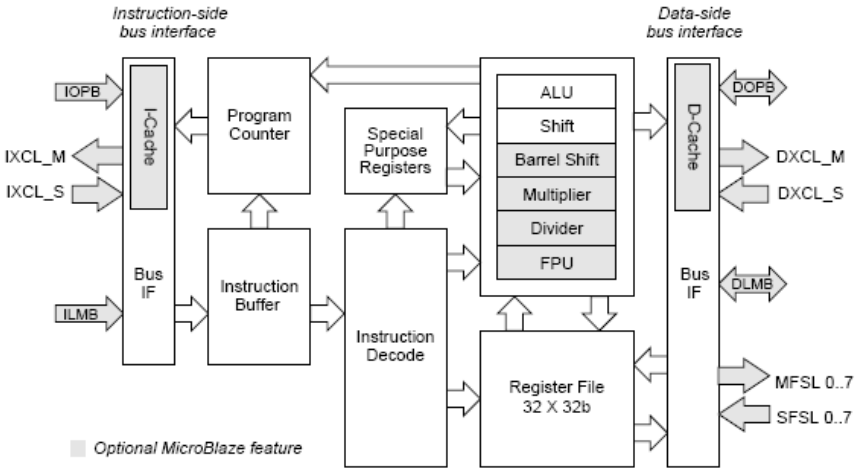

Figure 5.47. *Microblaze processor core block diagram. Figure taken from [101].*

In this design, 32 KB of data and instruction memory have been used. These memories have been implemented using the FPGA block RAM (BRAM) resources [89, 101]. Two LMBs have been used to access these memories. The LMB is a synchronous bus used primarily to access on-chip block RAM. It uses a minimum number of control signals and a simple protocol to ensure that local block RAM are accessed in a single clock cycle.

Also a data OPB (DOPB) has been used for connecting the SDRAM controller peripheral to the processor core. An OPB system is composed of masters, slaves, a bus interconnect and an arbiter. The bus interconnect is a distributed multiplexer implemented as an AND function in the master or slave driving the bus, and an OR function to combine the drivers into a single bus. The OPB bus signals are created by a logical OR of the signals that drive the bus. OPB devices that are not active during a transaction are required to drive zeros into the OR structure. This structure forms a distributed multiplexer and results in efficient bus implementations in FPGAs. Bus arbitration signals are directly connected between the OPB arbiter and each OPB master device. The *Microblaze* DOPB interface is organized as byte-

enable capable only masters while the SDRAM controller peripheral will act as slave. The arbiter and the bus interconnect is implemented in the OPB_v20 [104] peripheral used in this design.

The SDRAM controller peripheral will implement the logic for writing, reading and controlling the external SDRAM. It has been implemented using the *OPB SDRAM Controller* module [105] which provides a SDRAM controller that connects to the OPB and the control interface for SDRAMs. It has been configured for a 8 bits data width bus and for a 256 Mb SDRAM external memory. The access to the SDRAM external memory is very easy by using this SDRAM controller, the SDRAM is simply addressed from the *Microblaze*. In the figure 5.48 a block diagram of this SDRAM controller is shown.

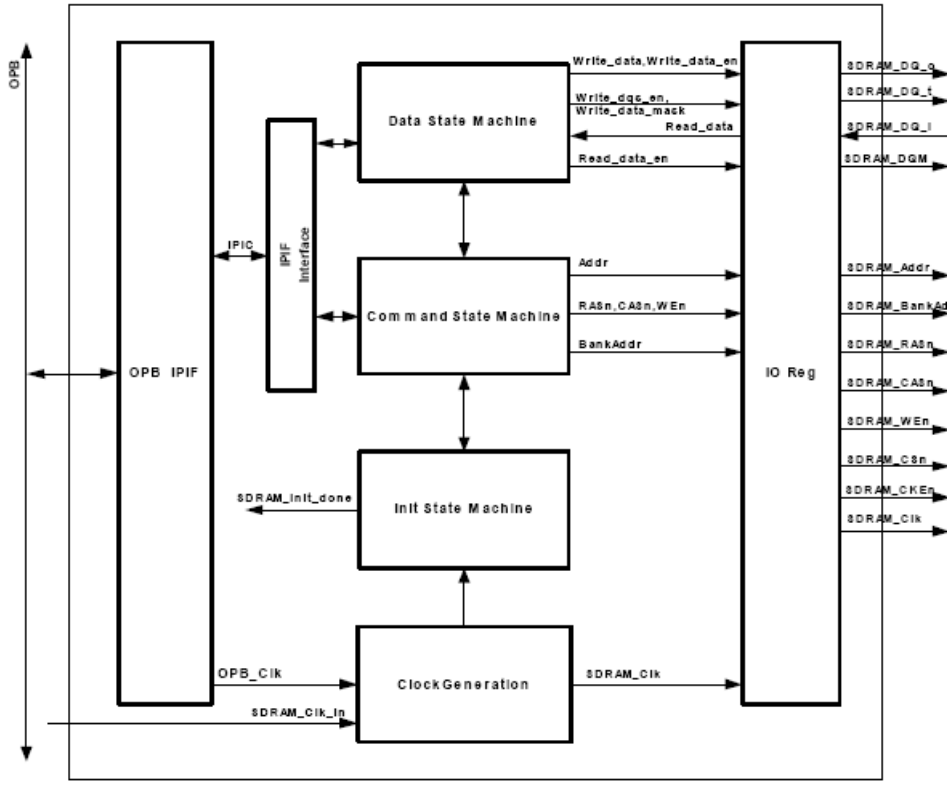

Figure 5.48. *OPB SDRAM Controller block diagram. Figure taken from [105].*

Six *Fast Simplex Links* (FSL), four SFSL (*Slave Fast Simplex Link*) and two MFSL (*Master FSL*), are used for very fast communication between the processor and the custom logic blocks. An Arbiter block which will be described in the next

section is also used for this purpose. The FSL [106] is a unidirectional point-to-point communication channel bus used to perform fast communication between any two elements on the FPGA when implementing an interface to the FSL bus.

The FSLs in this design have been implemented as asynchronous FIFOs (*i.e.* different clock signals at the input and at the ouput of the FIFO) using for their implementation LUTs (*Look Up Tables*) instead of BRAMs [89]. Four FSLs have been implemented as slaves (*i.e.* the data will be read by the processor),

- ADC_input_0: asynchronous FIFO of 128 by 32 bits where data are written by one *ADC Control* block and the data can be read one clock cycle after by the processor. It is connected directly to one *ADC Control* block trough the *Arbiter* block.
- ADC_input_1: asynchronous FIFO of 128 by 32 bits where data are written by the other *ADC Control* block and the data can be read one clock cycle after by the processor. It is connected directly to the other *ADC Control* block trough the *Arbiter* block.
- USB_input: asynchronous FIFO of 16 by 32 bits where data are written by the *USB Control* block and the data can be read one clock cycle after by the processor. It is connected directly to the *USB Control* block trough the *Arbiter* block.
- Data_input: asynchronous FIFO of 16 by 32 bits where data are written by the *Arbiter* block and the data can be read one clock cycle after by the processor. It is connected to the *Arbiter* block for reading data from the *Arbiter* block registers which correspond to the output signals of the custom logic blocks.

Two FSLs have been implemented as masters (*i.e.* the data will be written by the processor):

- USB_output: asynchronous FIFO of 16 by 32 bits where data are written by the processor and the data can be read one clock cycle after by *USB Control* block. It is connected directly to the *USB Control* block trough the *Arbiter* block.
- Data_output: Asynchronous FIFO of 16 by 32 bits where data are written by the processor and the data can be read one clock cycle after by *Arbiter* block. It is connected to the *Arbiter* block for writing data to the *Arbiter* block registers which correspond to the input signals of the custom logic blocks.

The access to these FSLs from the processor is carried out with a non-addressed

mode, special instructions are available for this access. The block diagram of the logic implemented in the FPGA including the embedded system and the Arbiter block can be seen in figure 5.49.

5.3.12. *Arbiter* block

Additional custom logic is needed in order to connect the signals of the custom logic blocks, both inputs and outputs, with the embedded system by means of six FSLs. This custom logic has been implemented in the *Arbiter* block. This block groups all the connections between the internal signals of the custom logic blocks (all of them described previously for each custom logic block as the signals connected to the *CFSM*) and the FSLs of the embedded system. These connections have been carried out either by means of direct connections or by means of a number of registers. There are some connections which have been implemented as direct connections between the custom logic blocks and the FSLs of the embedded system. These direct connections are the following,

- ADC_input_0: this input FIFO (from the embedded system point of view) is driven directly by one *ADC Control* block. Thus, for each acquisition the acquired readout (128 x 16 bits) is written in this FIFO by the corresponding *ADC Control* block and this readout can be read directly by the *Microblaze* processor one clock cycle later. Since this input FIFO has a size of 128 x 32 bits and the readout has a size of 128 x 16 bits, each 16 bits data is filled in the *Arbiter* with other 16 bits (all zeros) to complete the FIFO width.
- ADC_input_1: this input FIFO (from the embedded system point of view) is driven directly by the other *ADC Control* block. All that it has been said for the ADC-input_0 applies to the ADC_input_1.
- USB_input: this input FIFO (from the embedded system point of view) is driven directly by the *USB Control* block. Therefore, data which have been sent from the host computer software to the mother board are read by the *USB Control* block and they are written in this FIFO. Then, these data can be read one clock cycle later by the embedded processor. Since this input FIFO has a size of 16 x 32 bits and USB data have a size of 1 x 8 bits, these 8-bit data are filled in the *Arbiter* with other 24 bits (all zeros) to complete the FIFO width.
- USB_output: This output FIFO (from the embedded system point of view) is connected directly by the *USB Control* block. Thus, data which must be sent from the mother board to the host computer software are written in this FIFO and one clock cycle later it can be read by the USB Control block, which will send the data to the host computer software by means of

the USB controller. The *Arbiter* block selects the 8 bits less significant of the output of this FIFO, since the FIFO size is 16 x32 bits and the *USB Control* block bus data width is 8 bits.

The rest of connections of the custom logic blocks with the embedded system have been implemented by means of registers and two FSLs, one for the communication from the embedded system to the custom logic blocks and another one for the communication from the custom logic blocks to the embedded system. The following FSLs are used for this communication,

- Data_input: this is an input FIFO (from the embedded system point of view) of 16 x 32 bits. Each 32-bit data word written by the *Arbiter* block can be read one clock cycle later by the embedded processor.
- Data_output: this is an output FIFO (from the embedded system point of view) of 16 x 32 bits. Each 32-bit data word written by the embedded processor can be read one clock cycle later by the *Arbiter* block.

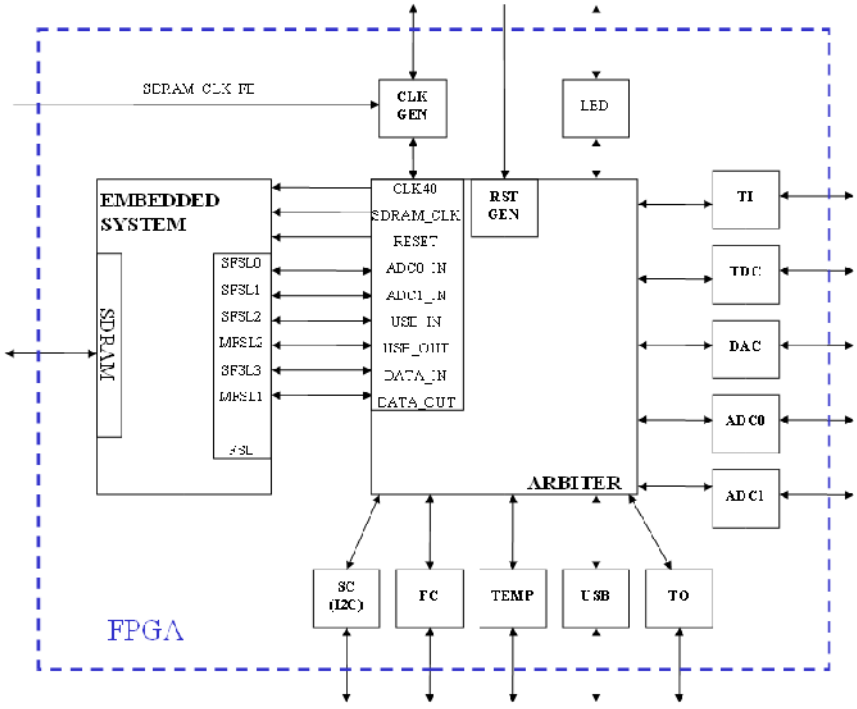

Figure 5.49. *Block diagram of the logic implemented in the system FPGA including the Embedded system and the Arbiter block.*

The registers implemented in the *Arbiter* are summarized in the Table 5.6. There

are eight registers of 32 bits each one. Five of these 32-bit registers are output registers from the point of view of the embedded system (*i.e.* they are written by the embedded processor and the output signals of these registers drive the inputs of the custom logic blocks). The other three registers are input registers regarding the embedded system.

The registers are addressed by the embedded processor by means of the three more significant bits written on the *Data_Output* FSL. Therefore, there are two different operations for addressing the registers depending on the type of register. For writing the write-only registers (RESET, CONTROL1, CONTROL2, CONTROL3 and CONTROL4) the embedded processor must write the address of the register and the data to be written on the *Data_Out* FSL. One clock cycle later, the *Arbiter* block will read the data and will write on the corresponding write-only register taking into account the 3-bit address. For this reason the three more significant bits of the write-only registers are reserved for the register address. For reading the read-only registers (STATUS, TEMP and TIME) the embedded processor must write only the address of the corresponding register on the the *Data_Out* FSL. One clock cycle later, the *Arbiter* block will read the address of the register and will select the corresponding register. One clock cycle later, the *Arbiter* block will write on the *Data_Input* FSL the contents of the previously addressed read-only register. Finally, one cycle later, the embedded processor will be able to read the written data on the *Data_Input* FSL by the *Arbiter* block.

Register Name	**Address**	**Register Description**	**Register Direction**
RESET	000	Reset signals for custom logic blocks	From embedded system to blocks

Register bit number

31-29	28-10	9	8	7	6	5	4	3	2	1	0
Address '000'	Not used	USB	ADC0	ADC1	Temp	SC	FC	TRO	TDC	DAC	LED

Register Name	**Address**	**Register Description**	**Register Direction**
CONTROL1	001	Control signals for custom logic blocks	From embedded system to blocks

Register bit number

31-29	28-25	24	23-19	18-11	10
Address '001'	Not used	Slow Control (SC)			Fast Control (FC)
		PROG	AD(4:0)	DATA(7:0)	CAL

9	8	7	6	5	4	3	2	1-0
ADC0			ADC1			Temp	LED	
WAKEUP	PD	ND_ACK	WAKEUP	PD	ND_ACK	ND_ACK	PROG	AD(1:0)

Register Name	**Address**	**Register Description**	**Register Direction**
CONTROL2	010	Control signals for custom logic blocks	From embedded system to blocks

Register bit number			
31-29	28-17	16	15-0
Address '010'	Not used	Trigger Out (TRO)	
		SAMPLE	NSAMPLES(15:0)

Register Name	Address	Register Description	Register Direction
CONTROL3	011	Control signals for custom logic blocks	From embedded system to blocks

Register bit number					
31-29	28-18	17	16-9	8	7-0
Address '011'	Not used	Trigger Out (TRO)			
		DCOARSE	COARSE_COUNT(7:0)	DFINE	FINE_COUNT(7:0)

Register Name	Address	Register Description	Register Direction
CONTROL4	100	Control signals for custom logic blocks	From embedded system to blocks

Register bit number										
31-29	28-21	20	19	18	17-16	15-4	3	2	1	0
Address '100'	Not used	Arbiter	USB	DAC			TDC		Trigger In	
		RST_T	ND_ACK	PROG	AD(1:0)	DATA(11:0)	ND_ACK	EN	CNIM	NE

Register Name	Address	Register Description	Register Direction
STATUS	101	Status signals from custom logic blocks	From blocks to embedded system

Register bit number										
31-22	21	20	19	18	17	16	15	14	13	12
Not used	SDRAM	FC		USB		SC		ADC0		
	INIT_DONE	TRIG	READY	READY	ND	READY	FAIL	FAIL	READY	SLEEP

11	10	9	8	7	6	5	4	3	2	1	0
ADC0	ADC1				Temp		TRO	DAC	TDC		
ND	FAIL	READY	SLEEP	ND	READY	ND	READY	READY	ND	READY	FAIL

Register Name	Address	Register Description	Register Direction
TEMP	110	Temperature data	From blocks to embedded system

Register bit number	
31-16	15-0
Not used	DTEMP(15:0)

Register Name	Address	Register Description	Register Direction
TIME	111	TDC time measurement	From blocks to embedded system

Register bit number
31-0
DTIME(15:0)

Table 5.6. *Summary of the Arbiter block registers.*

As well as the custom blocks signals and the embedded system signals, the *Arbiter* block have three input signals from the *Clock Generator*:

- CLK40: reference clock signal of 40 MHz used by all the FPGA blocks.
- RST: active high asynchronous reset signal for the block.
- CLK_SDRAM: clock signal of 40 MHz used by the SDRAM controller peripheral of the embedded system to drive the external SDRAM memory.

These signals are connected also to the embedded system through the Arbiter block in order to provide the same clock and reset signals to the embedded system than the rest of the FPGA logic for having a synchronous system.

5.3.13. *Clock Generator* block

In this design, all the blocks of the FPGA have sequential logic, that is, they need a clock signal to operate. Furthermore, since the blocks have to operate in a synchronous mode, the clock signal for all the blocks must be the same. For achieving this purpose the clock skew on the reference clock signal (in this case a clock signal with a frequency of 40 MHz, CLK40_0) should be eliminated.

As well as the common reference clock signal of 40 MHz, other clock signals must be generated for the two *ADC Control* blocks and for the SDRAM controller peripheral of the embedded system. The clock signals for the *ADC Control* blocks have been described in the section 5.3.1 of this chapter. Some of these signals (CLK40_0I, CLK40_90I, CLK40_180I and CLK40_270I) are routed in the FPGA using normal paths instead of specific clock paths and they use different internal buffers. This is done because these clock signals are used to generate the SCLK sampling clock signals for the ADCs by using combinational logic (*i.e.* they drive the inputs of combinational gates) and this would cause too much skew if the specific clock distribution resources were used to route them in the FPGA. The clock signal for the SDRAM controller peripheral is used by this controller to drive the external SDRAM. Therefore, the clock skew of this signal must be controlled independently since this signal will have also an external path.

The most of the FPGA blocks of this design also need a reset signal. This signal is generated by the *Clock Generator* from an external reset input signal (EXT_RST) which comes from the reset circuit of the mother board. This external signal is generated from a push button so it will have to be processed in this block to avoid the characteristic glitches as well as to generate a shorter pulsed internal reset signal (INT_RST) for the FPGA blocks. Moreover, this internal reset signal

must be active until all clock signals are stable.

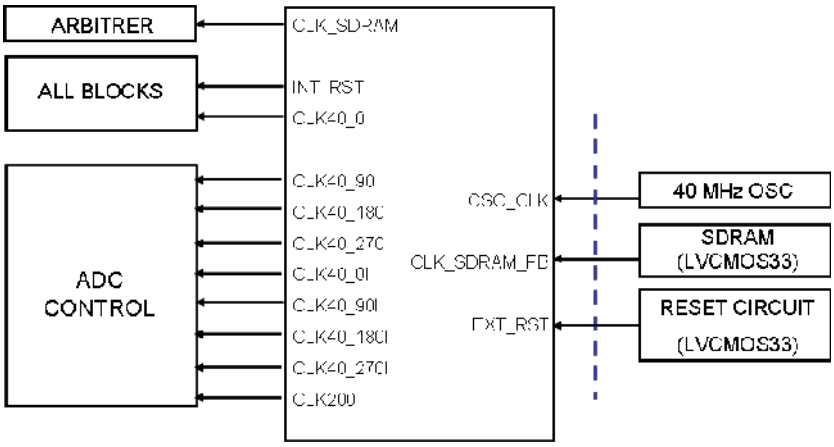

Figure 5.50. *Clock Generator block and its connections with external circuits and other internal FPGA blocks.*

The Clock Generator block is shown in the figure 5.50. The aforementioned requirements for this block are achieved by using the on-chip *Digital Clock Managers* (DCM). A DCM block diagram can be seen in the figure 5.51. A DCM [89, 107] integrates different clocking capabilities into the Spartan 3 FPGA global clock distribution network. Consequently, a DCM solves a variety of common clocking issues,

- multiplying or dividing an incoming clock frequency or synthesizing a completely new frequency by a mixture of clock multiplication and division.
- Conditioning a clock, ensuring a clean output clock with a 50% duty cycle.
- Phase shifting a clock signal, either by a fixed fraction of a clock period or by precise increments.
- Eliminating clock skew, either within the device or to external components, to improve overall system performance and to eliminate clock distribution delays.
- Mirroring, forwarding, or rebuffering a clock signal, often to deskew and convert the incoming clock signal to a different I/O standard.

Three of the four DCMs available in the FPGA are used in this block. The clock signals of this block are generated in FPGA from a 40 MHz input clock signal (OSC_CLK) produced by an external clock oscillator. The aim for using the DCM was to eliminate the skew on the different clock signals generated. The clock edge

generated by the clock source arrives at different times at different points in the system, either within a single device or on the clock inputs to the different devices connected to the clock. This difference in arrival times (Δb and Δc) is defined as clock skew, as can be seen in the figure 5.52 on the left side.

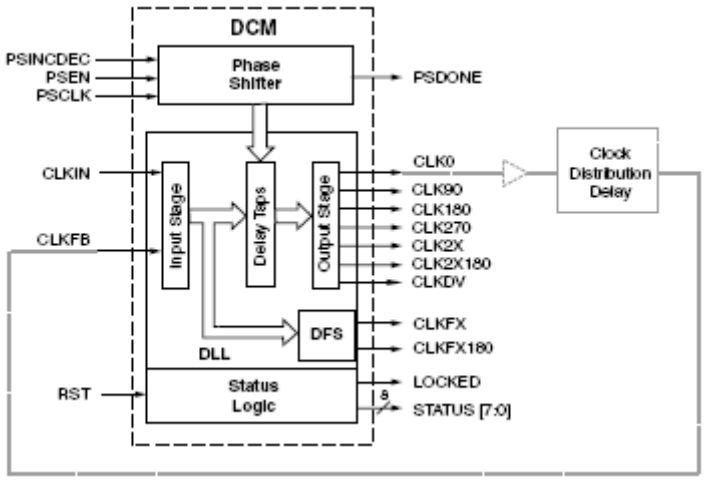

Figure 5.51. *DCM functional block diagram. Figure taken from [107].*

The elimination of the clock skew using DCMs in a Spartan 3 FPGA is represented in the figure 5.52 on the right side. Two DCMs eliminate the clock skew: One DCM eliminates the skew for clocked items within the FPGA, the other DCM eliminates the skew when clocking another device on the board. Because it is impossible to insert a negative delay to counteract the clock skew, the DCM delays the clocks enough so that they appear to be advanced in time. The clock cycle is repetitive and has a fixed period, T. The clock at point B appears to be advanced in time by the delay Δb. However, the clock is delayed by $T - \Delta b$. Similarly, the clock at point C is delayed by $T - \Delta c$.

Following this philosophy and the requirements for the generation of the different clock signals, the following DCMs scheme has been used,

- DCM0: this DCM is used for generating the clock signal for the SDRAM controller peripheral (CLK_SDRAM) which will drive with this signal the external SDRAM. A sample of the CLK_SDRAM clock signal is sent back to the FPGA (CLK_SDRAM_FB) and it is connected to the CLKFB of this DCM to eliminate the clock skew. Feedback path delay for CLK_SDRAM_FB must match the forward path delay for CLK_SDRAM

to guarantee skew elimination. The input clock signal for this DCM is the external oscillator signal (OSC_CLK).

- DCM1: its purpose is to generate the reference clock signal (CLK40_0) for all the logic blocks and the delayed clock signals for the *ADC Control* blocks (CLK40_90, CLK40_180 and CLK40_270). Also the signals used for generating the SCLK signals of the *ADC Control* blocks are derived (CLK40_0I, CLK40_90I, CLK40_180I and CLK40_270I) from previous clock signals (CLK40_0, CLK40_90, CLK40_180 and CLK40_270). A sample of the CLK40_0 clock signal is connected internally to the CLKFB of this DCM to eliminate the clock skew. The input clock signal for this DCM is the external oscillator signal (OSC_CLK).
- DCM2: this DCM generates the 200 MHz clock signal (CLK200) for the *ADC Control* blocks. A sample of the CLK200 clock signal is connected internally to the CLKFB of this DCM to eliminate the clock skew. The input clock signal for this DCM is the external oscillator signal (OSC_CLK).

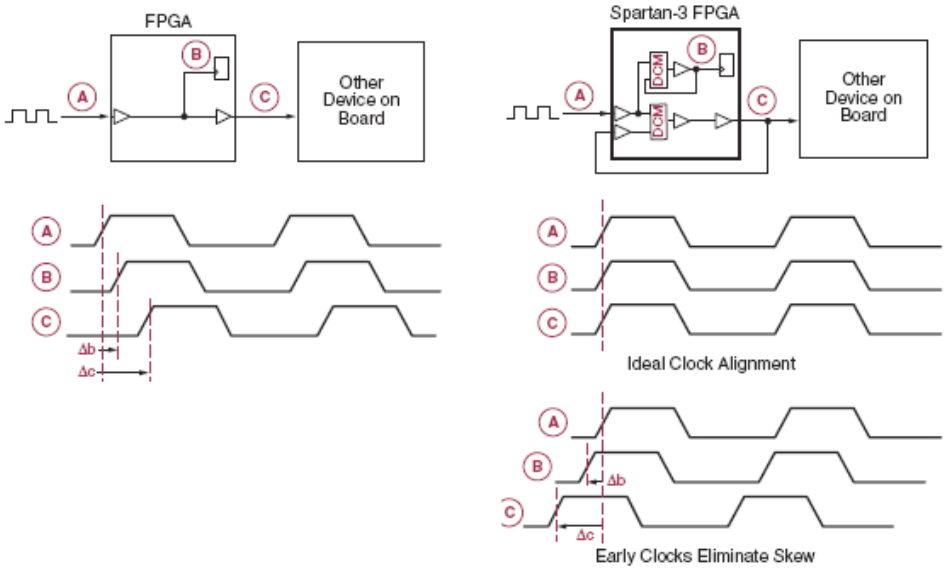

Figure 5.52. *Clock skew example in a FPGA system (left side). Clock skew elimination in a Spartan 3 FPGA using DCMs (right side). Figure taken from [107].*

Regarding the reset generation, custom logic has been designed in the block to generate an active high internal reset signal (INT_RST) from the unfiltered active low external reset signal (EXT_RST). The INT_RST signal will be active when EXT_RST will be active during a fixed time (at least 2 ms) and then a leading edge

on the EXT_RST is detected to prevent the bounces from the reset push button. Otherwise, a low level on EXT_RST will be ignored. The DCMs also are reset with this condition. Consequently, the INT_RST signal will be also active until the three DCMs have locked their output clock signals (*i.e.* the clock signals are stable).

5.3.14. FPGA resources utilization

In the table 5.7 there are summarized the FPGA resources consumed in this design by the implemented logic, including the embedded system. It can be seen that there is room to accommodate more logic if required.

FPGA xc3s400-5pq208 device utilization			
Logic utilization	**Used**	**Available**	**% Utilization**
Slices	3186	3584	88 %
Slice Flip-Flops	1780	7168	24 %
4 input LUTs	3671	7168	51 %
BRAMs	16	16	100 %
Mult 18x18	3	16	18 %

Table 5.7. *Summary of the FPGA resources consumed.*

5.4. FPGA firmware

The hardware functionality is controlled at low level by the mother board FPGA. In particular, the firmware programmed in the embedded processor is in charge of interpreting the host computer software orders sent by USB and executing these orders by means of the custom logic blocks. These custom logic blocks control the different hardware circuits of both the mother board and the daughter board. Furthermore, this firmware has to organize and send the acquired data by USB communication.

The embedded firmware has been designed as a finite state machine (FSM) whose states correspond to the different functions which have to be carried out by the system. The firmware has been programmed using C language [108]. The microblaze embedded processor offers some pre-designed functions and libraries to control the different peripherals used with the processor. In this case, functions for accessing the external SDRAM and the internal FSLs have been used. The figure 5.53 shows the possible states of the system, which corresponds to the states of the firmware FSM. The FSM has twelve different states. In the following sections,

each state is described with more detail.

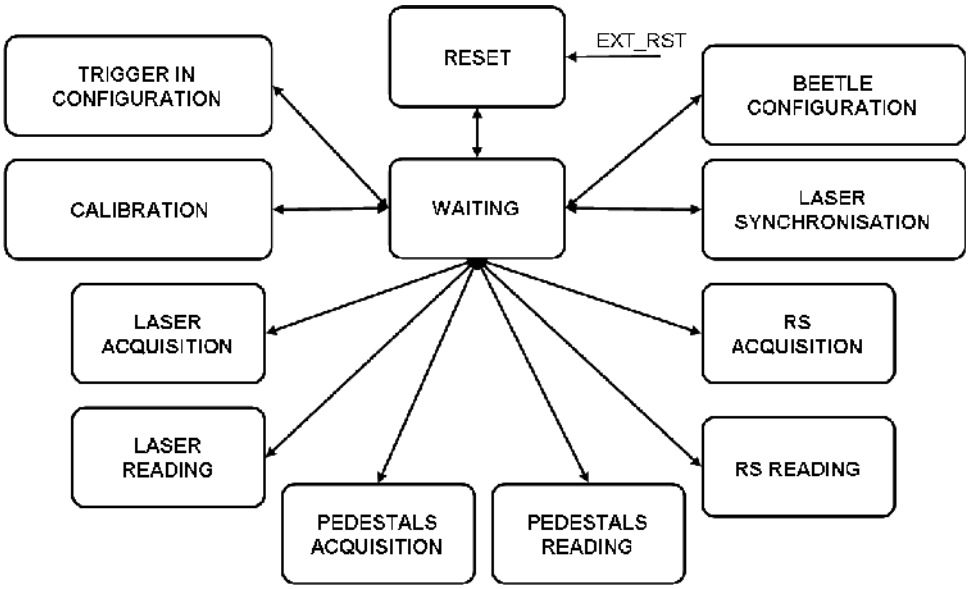

Figure 5.53. *States of the FSM implemented in the embedded processor firmware. These states are the possible states of the system.*

5.4.1. *Reset* state

The initialization of the custom logic blocks of the FPGA is performed in this state. The different data exchanged between the mother board and the host computer software while in this state are summarized in the table 5.8. After the initialization, the USB communication with the software is checked out by receiving a connection code from the software, by sending another code to the software (*Reset code*) and finally receiving an acknowledge code (*Reset ack*). This state can be accessed by means of an external reset or a software command (from the *Waiting* state). When the tasks of this state are finished, the system moves to the *Waiting* state. During the *Reset* state, both motherboard LEDs are on.

Code/Data Name	Number of bits	Format/Description (MSB.....LSB)	Direction
Connection code	8 bits	0x0F (Hex)	SW-MB
Reset code	8 bits	0x01 (Hex)	MB-SW
Reset ack	8 bits	0x01 (Hex)	SW-MB

Table 5.8. *Summary of the data exchanged between the mother board (MB) and the software (SW) in the Reset state.*

5.4.2. *Waiting* state

Once the system has been initialized and synchronized with the host computer software during the *Reset* state, the system remains in this state waiting for new orders from the host computer software. When this state is accessed, the mother board sends a *Waiting code* and only the green LED will be on. Then, the system waits for a code from the software which indicates the following state which must be accessed by the system (*following state code*). The different data which can be exchanged between the mother board and the software in this state are summarized in the table 5.9.

Code/Data Name	Number of bits	Format/Description (MSB.....LSB)	Direction
Waiting code	8 bits	0x02	MB-SW
Following state code: following state to access			
Reset	8 bits	0x01	SW-MB
Beetle Configuration		0x03	
Calibration		0x04	
Trigger In Configuration		0x05	
Laser Synchronization		0x06	
Laser Acquisition		0x07	
Laser Reading		0x08	
RS Acquisition		0x09	
RS Reading		0x0A	
Connection		0x0F	

Table 5.9. *Summary of the data exchanged between the mother board (MB) and the software (SW) in the Waiting state.*

5.4.3. *Beetle Configuration* state

The configuration of the Beetle chips is carried out in this state by means of the *Slow Control* block. This state must be the first state accessed after the system has been in the *Waiting* state for the first time (*i.e.* after the *Reset* state). In first place, a *Beetle Configuration code* is sent to the software. Then, a *New data code* is sent to the software demanding new configuration data or a *End of data code*.

The internal registers of the Beetle chips are configured according to the configuration data received from the software by the *Slow Control* block. Then the data programmed in the Beetle chips registers are read back and checked out. If a failure is detected in the *Slow Control* block (either by a data check error or by an

I2C communication error), a *Slow Control error code* is sent to the software, the red LED of the mother board is turned on (and the green LED turned off) and the system goes to the *Waiting* state. If there is no error in the last programmed data, the *New data code* is sent to the software again demanding new configuration data or the *End of data code*. When the *End of data code* is received (*i.e.* after configuring correctly all the Beetle chips internal registers) the system exits the loop and comes back to the *Waiting* state. The different data which can be exchanged between the mother board and the software in this state are summarized in the table 5.10.

Code/Data Name	Number of bits	Format/Description (MSB.....LSB)	Direction
Beetle Configuration code	8 bits	0x03	MB-SW
New data code	8 bits	0x10	MB-SW
Configuration data [0 0 0 AD(4:0) DATA(7:0)]			
Register 0 (Itp)		0x0000	
Register 1 (Ipre)		0x014C	
Register 2(Isha)		0x020A	
Register 3 (Ibuf)		0x030A	
Register 4 (Vfp)		0x0400	
Register 5 (Vfs)		0x0500	
Register 9 (Vrc)		0x0900	
Register 10 (Ipipe)		0x0A0D	
Register 12 (Vd)	16 bits	0x0B82	SW-MB
Register 12 (Vdcl)		0x0C69	
Register 13 (Ivoltbuf)		0x0D14	
Register 14 (Isf)		0x0E1A	
Register 15 (Icurrbuf)		0x0F66	
Register 16 (Latency)		0x1080	
Register 17 (ROCtrl)		0x111A	
Register 18 (RclkDiv)		0x1200	
Register 19 (CompCtrl)		0x1309	
End of data code	16 bits	0x8000	SW-MB
Slow Control error code	8 bits	0x20	MB-SW

Table 5.10. *Summary of the data exchanged between the mother board (MB) and the software (SW) in the Beetle Configuration state.*

5.4.4. *Trigger Input Configuration* state

The *Trigger Input* block (*i.e.* which trigger inputs are going to be used and if a

coincidence is going to be applied) as well as the DAC discrimination thresholds are configured in this state. When the system enters in this state a *Trigger In Configuration code* is sent to the software. Then, *Trigger In Configuration data* are received from the software and the *Trigger Input* block is configured. Then, a *New data code* is sent to the software demanding new DAC configuration data or the *End of data code*. If DAC configuration data are received the corresponding DAC threshold is programmed and a *New data code* is sent to the software again. When the *End of data code* is received (*i.e.* after configuring correctly all the DAC thresholds) the system exits the loop and comes back to the *Waiting* state. The different data which can be exchanged between the mother board and the software in this state are summarized in the table 5.11.

Code/Data Name	Number of bits	Format/Description (MSB.....LSB)	Direction
Trigger In Configuration code	8 bits	0x05	MB-SW
Tigger In Configuration data	8 bits	[0...0 CNIM NE]	SW-MB
New data code	8 bits	0x10	MB-SW
DAC configuration data [0 0 AD(1:0) DATA(11:0)]			
DAC A (Trig In 1)	16 bits	[0 0 0 0 DATA(11:0)]	SW-MB
DAC B (Trig In 2)		[0 0 0 1 DATA(11:0)]	
DAC C (Trig Pulse In)		[0 0 1 0 DATA(11:0)]	
DAC D (Trig Pulse In)		[0 0 1 1 DATA(11:0)]	
End of data code		0x8000	
End of data code	16 bits	0x8000	SW-MB

Table 5.11. *Summary of the data exchanged between the mother board (MB) and the software (SW) in the Trigger Input Configuration state.*

5.4.5. *Calibration* state

In this state, the system acquires the calibration data by programming the Beetle chips to generate input pulses according to a specific input charge and then acquiring the data corresponding to each input pulse generated previously. This state must be accessed after the system has been in the *Beetle Configuration* state (*i.e.* the Beetle chips have been configured and initialized). When this state is accessed a *Calibration code* is sent to the software. Then, the Beetle chips are configured to start a calibration and an error condition is checked in the *Slow Control* block. If so, a *Slow Control error code* is sent to the software, the red LED of the mother board is activated and the system comes back to the *Waiting* state.

Otherwise, a *New data code* is sent to the software in order to demand new calibration data or the *End of data code*. If *Calibration data* are received, the

Beetle chips are configured according to these data (*i.e.* a specific charge to generate the corresponding input calibration pulse at the Beetle chips) and an error condition is checked in the *Slow Control* block. If so, a *Slow Control error code* is sent to the software, the red LED of the mother board is activated and the system comes back to the *Waiting* state. If there is no error, the *Fast Control* block is programmed to generate the input calibration pulse at the Beetle chips.

Code/Data Name	Number of bits	Format/Description (MSB.....LSB)	Direction
Calibration code	8 bits	0x04	MB-SW
New data code	8 bits	0x10	MB-SW
Calibration data [0 0 0 AD(4:0) DATA(7:0)]			
Register 0 (Itp): 0 e^-	16 bits	0x0000	SW-MB
Register 0 (Itp): 1025 e^-		0x0001	
...		...	
Register 0 (Itp): 261375 e^-		0x00FF	
Acquired data from calibration [Canal 0 (16 bits)...Canal 255 (16 bits)]			
Calibration data 1 (0 e^-)	256x16 bits	[Ch 0 (16 bits)...Ch 255 (16 bits)]	MB-SW
Calibration data 2 (±1025 e^-)		[Ch 0 (16 bits)...Ch 255 (16 bits)]	
...		...	
Calibration data 256 (±261375 e^-)		[Ch 0 (16 bits)...Ch 255 (16 bits)]	
End of data code	16 bits	0x8000	SW-MB
Slow Control error code	8 bits	0x20	MB-SW
ADC Control 0 error code	8 bits	0x30	MB-SW
ADC Control 1 error code	8 bits	0x40	MB-SW
ADC Control 0 /ADC Control 1 error code	8 bits	0x50	MB-SW

Table 5.12. *Summary of the data exchanged between the mother board (MB) and the software (SW) in the Calibration state.*

Then, the corresponding data are acquired. During this process an error condition is checked in both *ADC Control* blocks. If so, the corresponding *ADC Control error code* is sent to the software, the red LED of the mother board is activated and the system comes back to the *Waiting* state. Otherwise, the data acquired from the Beetle chips are sent to the software and a *New data code* is sent to the software again demanding new calibration data or the *End of data code*. When the *End of data code* is received (*i.e.* after acquiring the required calibration data) the system exits the loop.

Then, the Beetle chips are configured to finish the calibration and an error

condition is checked in the *Slow Control* block. If so, a *Slow Control error code* is sent to the software, the red LED of the mother board is activated and the system comes back to the *Waiting* state. Else, the system comes back to the *Waiting* state. The different data which can be exchanged between the mother board and the software in this state is summarized in the table 5.12.

5.4.7. *Laser Synchronization* state

In this state, the delay between the TRIG OUT pulse signal (driving the laser source) and the *fast control Trigger* signal (triggering the Beetle chips data readout) is synchronized. This process is controlled by the user through the software. Prior to this state, the system has had to access the *Beetle Configuration* state.

Code/Data Name	Number of bits	Format/Description (MSB.....LSB)	Direction
Laser Synchronization code	8 bits	0x06	MB-SW
New data code	8 bits	0x10	MB-SW
Synchronization data [delay in ns (0-6375 ns)]			
Synchronization data 1 Synchronization data 2 ... Synchronization data n	16 bits	[delay in ns (0 -6375 ns)]	SW-MB
Acquired data from synchronization state [Channel 0 (16 bits)... Channel 255 (16 bits)]			
Synchronization data 1 Synchronization data 2 ... Synchronization data n	256x16 bits	[Ch 0 (16 bits)... Ch (16 bits)] [Ch 0 (16 bits)... Ch (16 bits)] ... [Ch 0 (16 bits)... Ch (16 bits)]	MB-SW
End of data code	16 bits	0x8000	SW-MB
ADC Control 0 error code	8 bits	0x30	MB-SW
ADC Control 1 error code	8 bits	0x40	MB-SW
ADC Control 0 /ADC Control 1 error code	8 bits	0x50	MB-SW

Table 5.13. *Summary of the data exchanged between the mother board (MB) and the software (SW) in the Laser Synchronization state.*

When this state is accessed a *Laser Synchronization code* is sent to the software. Then, a *New data code* is sent to the software in order to demand new synchronization data or a *End of data code*. The *Trigger Output* block is programmed with the synchronization data received. Then, the Beetle chips data corresponding to the last synchronization data programmed are acquired. During

this process an error condition is checked in both *ADC Control* blocks. If so, the corresponding *ADC Control error code* is sent to the software, the red LED of the mother board is activated and the system comes back to the *Waiting* state. Otherwise, the data acquired from the Beetle chips are sent to the software and a *New data code* is sent to the software again demanding new synchronization data or the *End of data code*.

When the *End of data code* is received (*i.e.* after synchronizing the system for the laser setup) the system exits the loop and comes back to the *Waiting* state. The different data which can be exchanged between the mother board and the software in this state is summarized in the table 5.13.

5.4.8. *Acquisition* states (*Laser*, *Radioactive Source* and *Pedestals*)

The acquisition states have been implemented to acquire data with the laser setup, the radioactive source setup or just the pedestals (*i.e.* data corresponding to zero input collected charge at the Beetle chips input).

Code/Data Name	Number of bits	Format/Description (MSB.....LSB)	Direction
Laser Acquisition code	8 bits	0x07	MB-SW
Laser Acquisition configuration data	16 bits	[LNSAMPLES(15:0)]	SW-MB
ADC Control 0 error code	8 bits	0x30	MB-SW
ADC Control 1 error code	8 bits	0x40	MB-SW
ADC Control 0 /ADC Control 1 error code	8 bits	0x50	MB-SW
Laser Acquisition error code	8 bits	0x60	MB-SW
SDRAM error code	8 bits	0x80	MB-SW

Table 5.14. *Summary of the data exchanged between the mother board (MB) and the software (SW) in the Laser Acquisition state.*

In the case of the *Laser Acquisition* state, a programmable number of samples (up to 64776) of a specific point of the Beetle chips front-end pulse shape, according to the synchronization carried out previously, is acquired using the laser setup. With each sample a temperature data and the analogue data corresponding to the 256 channels of the Beetle chips are acquired. These data are stored in the mother board SDRAM. Before this state, the system has had to access the *Beetle Configuration* state and the *Laser Synchronization* state. The different data which

can be exchanged between the mother board and the software in this state is summarized in the table 5.14.

Code/Data Name	Number of bits	Format/Description (MSB.....LSB)	Direction
RS Acquisition code	8 bits	0x09	MB-SW
RS Acquisition configuration data	16 bits	[RSNSAMPLES(15:0)]	SW-MB
TDC Control error code	8 bits	0x70	MB-SW
ADC Control 0 error code	8 bits	0x30	MB-SW
ADC Control 1 error code	8 bits	0x40	MB-SW
ADC Control 0 /ADC Control 1 error code	8 bits	0x50	MB-SW
SDRAM error code	8 bits	0x80	MB-SW

Table 5.15. *Summary of the data exchanged between the mother board (MB) and the software (SW) in the Radioactive Source Acquisition state.*

Code/Data Name	Number of bits	Format/Description (MSB.....LSB)	Direction
Pedestals Acquisition code	8 bits	0x0B (Hex)	MB-SW
Pedestals Acquisition configuration data	16 bits	[PDNSAMPLES(15:0)]	SW-MB
ADC Control 0 error code	8 bits	0x30 (Hex)	MB-SW
ADC Control 1 error code	8 bits	0x40 (Hex)	MB-SW
ADC Control 0 /ADC Control 1 error code	8 bits	0x50 (Hex)	MB-SW
SDRAM error code	8 bits	0x80 (Hex)	MB-SW
Pedestals Acquisition code	8 bits	0x0B (Hex)	MB-SW

Table 5.16. *Summary of the data exchanged between the mother board (MB) and the software (SW) in the Pedestals Acquisition state.*

In the case of the *Radioactive Source Acquisition* state, a programmable number of samples (up to 64776) are acquired using as trigger an input signal processed from the trigger input signals. For each sample a time measurement of the TDC, temperature data and the analogue data corresponding to the 256 channels of the Beetle chips are acquired. These data are stored in the mother board SDRAM. Before this state, the system has had to access the *Beetle Configuration* state and the *Trigger In Configuration* state. The different data which can be exchanged between the mother board and the software in this state is summarized in the table 5.15.

In the case of the *Pedestals Acquisition* state, a programmable number of

samples (up to 64776) is acquired from a generated internal trigger in order to obtain the pedestals of the Beetle chips output analogue signal. With each sample temperature data and the analogue data corresponding to the 256 channels of the Beetle chips are acquired. These data are stored in the mother board SDRAM. Before this state, the system has had to access the *Beetle Configuration* state. The different data which can be exchanged between the mother board and the software in this state is summarized in the table 5.16.

5.4.9. *Reading* states (*Laser*, *Radioactive Source* and *Pedestals*)

The reading states have been implemented to read the data acquired in the different acquisition states. In these states, the last data acquired in the corresponding acquisition state are read from the mother board SDRAM and they are sent to the software. All the states have the same structure. The different data which can be exchanged between the mother board and the software for the *Laser Reading* state, the *Radioactive Source Reading* state and the *Pedestals Reading* state is summarized in the tables 5.17, 5.18 and 5.19 respectively.

Code/Data Name	**Number of bits**	**Format/Description (MSB.....LSB)**	**Direction**
Laser Reading code	8 bits	0x08	MB-SW
Number of samples data (1 to 64776)	16 bits	[LSDRAMDATA(15:0)]	MB-SW
Number of bits by sample data	16 bits	[LDATAFRAMENBITS(15:0)]	MB-SW
Laser Reading ack	8 bits	0x08	SW-MB
Laser Acquisition stored data [Time(32 bits) Temperature (16 bits) Ch0 (16 bits)...Ch 255 (16 bits)]			
Sample 1 Sample 2 ... Sample n (n = 1...64776)	259x16 bits	Laser Acquisition data	MB-SW
SDRAM error code	8 bits	0x80	MB-SW

Table 5.17. *Summary of the data exchanged between the mother board (MB) and the software (SW) in the Laser Reading state.*

When any of the reading states is accessed, the corresponding state code is sent to the software. Then, the *Number of samples data* (1 to 64776) and the *Number of bits by sample data* (259 x 16 bits or 4144 bits) are also sent to the software. Afterwards, the system waits for receiving the corresponding acknowledge code from the software.

Code/Data Name	Number of bits	Format/Description (MSB.....LSB)	Direction
RS Reading code	8 bits	0x0A	MB-SW
Number of samples data (1 to 64776)	16 bits	[RSSDRAMDATA(15:0)]	MB-SW
Number of bits by sample data	16 bits	[RSDATAFRAMENBITS(15:0)]	MB-SW
RS Reading ack	8 bits	0x0A	SW-MB
RS Acquisition stored data [Time(32 bits) Temperature (16 bits) Ch 0 (16 bits)...Ch 255 (16 bits)]			
Sample 1 Sample 2 ... Sample n (n = 1...64776)	259x16 bits	RS Acquisition data	MB-SW
SDRAM error code	8 bits	0x80	MB-SW

Table 5.18. *Summary of the data exchanged between the mother board (MB) and the software (SW) in the Radioactive Source Reading state.*

Code/Data Name	Number of bits	Format/Description (MSB.....LSB)	Direction
Pedestals Reading code	8 bits	0x0C	MB-SW
Number of samples data (1 to 64776)	16 bits	[PDSDRAMDATA(15:0)]	MB-SW
Number of bits by sample data	16 bits	[PDDATAFRAMENBITS(15:0)]	MB-SW
Pedestals Reading ack	8 bits	0x0C	SW-MB
Pedestals Acquisition stored data [Time(32 bits) Temperature (16 bits) Ch 0 (16 bits)...Ch 255 (16 bits)]			
Sample 1 Sample 2 ... Sample n (n = 1...64776)	259x16 bits	Pedestals Acquisition data	MB-SW
SDRAM error code	8 bits	0x80	MB-SW

Table 5.19. *Summary of the data exchanged between the mother board (MB) and the software (SW) in the Pedestals Reading state.*

Then, the data stored in the SDRAM are read by the mother board and they are sent to the software. During this process an error condition is checked in the SDRAM controller. If so, the *SDRAM error code* is sent to the software, the red LED of the mother board is activated and the system comes back to the *Waiting* state. Otherwise, when the all the data stored in the SDRAM have been shipped the system will finish the loop and it will come back to the *Waiting* state.

Chapter 6

The software

In this chapter the software part of the system is described. The functions and the structure of the software are explained in section 6.1. According to this structure, the two main parts of the software are treated subsequently. In section 6.2, the low level software is described, which mainly is in charge of controlling the hardware by USB communication. In section 6.3, it is dealt with the high level software which includes the Graphical User Interface (GUI) for the communication between the user and the system. It also has the output file generation module for the generation of files containing the data acquired which can be stored in the host computer for further analysis.

6.1. Software structure

The main function of the software is to control the whole system and to process the data acquired from the sensors, by means of the hardware part of the system, as well as storing these data with an adequate format in the host computer for further analysis.

As it has been explained in chapter 5, the software controls the hardware part of the system by means of codes which are sent by USB communication and are interpreted by the FPGA logic of the mother board. Furthermore, the mother board FPGA also sends to the software the data acquired by the hardware and status codes informing about the state of the hardware. Thus, the software can control the hardware for configuration, calibration and data acquisition. Moreover, the software processes the data acquired (temperature data, TDC data and digitalized

Beetle chips data output), which are sent in a raw format from the mother board, in order to obtain data with physical meaning. Also, data introduced by the user are processed by the software to be sent to the FPGA in the correct format. This low level data processing is carried out with the software instead of the FPGA to take advantage of the host computer processing capacity, thus releasing FPGA resources for controlling the hardware and for acquiring the data.

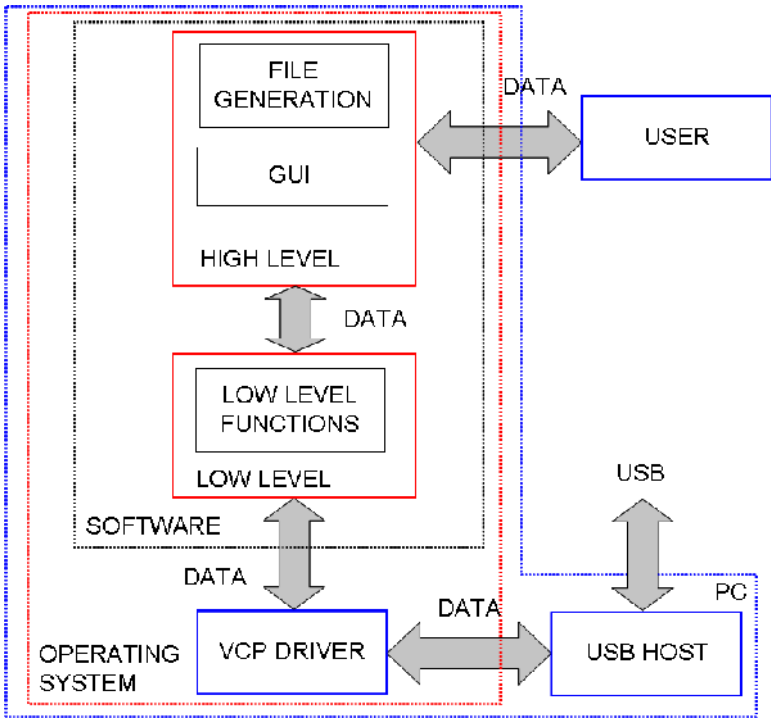

Figure 6.1. *Conceptual block diagram of the software structure.*

On the other hand, the software has to perform as interface between the user and the hardware part in order to configure and to control the system according to the user preferences as well as to monitor the state of the system and the data acquired for user information. This interface has been implemented as a *Graphical User Interface* (GUI). Finally, the software has to store the data acquired and processed at low level by generating an output data file which could be used with other software for further data analysis (statistic, plots, etc) like the ROOT framework [109].

Taking into account the above considerations, the software has been implemented by dividing it into two conceptual levels (figure 6.1), a low level

software and a high level software. The low level software deals with the data exchange by USB with the mother board and with the low level data processing. The high level software implements the GUI for the communication between the user and the system, the data monitoring as well as the data output file generation. These two levels will be described in the following sections. The software has been designed initially for running in a Linux operating system. Both the low level software and the high level software have been implemented in the same package and they have been designed using the C++ language [110]. A more detailed description of the software can be found on [111].

6.2. Low level software

The low level software is in charge of the USB communication with the mother board as well as the low level data processing. The USB communication is carried out by means of different hardware circuits and a driver on the host computer side. First, the mother board has a USB controller device which implements the required functions to read or write 8-bit data words from the FPGA and translate these data into USB format. This USB controller is interfaced by a FPGA logic block which manages the data flow. On the other hand, there is a USB host controller at the computer which is in communication with the mother board USB controller. This USB host controller needs a driver to be operated from the software. The design of this driver is not trivial, in fact this task is quite time consuming.

However, in this case the mother board USB controller manufacturer provides different types of drivers already tested in different operating systems to be used directly with the USB controller. Two different types of drivers have been considered for this design,

- VCP driver (*Virtual Com Port*): This driver emulates a conventional serial port. Thus, the software is designed as it was operating with a serial port. The driver deals with the translation for the communication with the USB host controller. The maximum data rate that can be achieved with this driver would be 2.4 Mbits/s.
- D2XX driver (*Direct driver*): This driver incorporates a library which works directly with the USB host controller. Therefore, in this case the software should use the functions available in this library to implement the communication with the PC USB host controller. The maximum data rate that can be achieved with this driver would be 8 Mbits/s.

Finally, the VCP driver option was selected since the required data rate for this system is not crucial and the software design is easier.

Regarding the data exchange between the hardware and the software, the low level software implements different objects to deal with all this data exchange. The data exchange between the FPGA firmware and the software has been described in the last section of the fifth chapter. Therefore, these software objects are used depending on the state of the FPGA firmware to generate the control codes required. Furthermore, these software objects also receive the status codes generated by the FPGA firmware, they interpret these codes and they execute different actions according to the state of the system and the user preferences. Some of these objects also receive the acquired data by the hardware and they process these data at low level. They also receive the data introduced by the user to configure the system and they process these data to be understood by the FPGA firmware. In particular, the data processed at low level are the following,

- ADC data: the ADC data sent by the FPGA firmware are composed of a 16 bits word for each of the 256 Beetle chips channels (256 x 16 bits for each sample acquired). For 16 bits word the most significant six bits are zeros and the remaining ten bits, DATA(9:0) corresponds to the digitalized value of the channel, codified as a ten bits signed number. The low level software processes the data of each channel by subtracting the corresponding average pedestal value for each channel. Therefore, a value for each channel in ADC counts (it could be negative) is obtained taking into account the pedestal value of that channel.

- TDC data: each TDC measurement is a 32 bits fixed point number, TIME(31:0) with the most significant 16 bits corresponding to the integer part (signed number) and the second 16 bits to the fractional part, representing a number of times (up to ±1.99) of a calibration clock period (100 ns). The fractional part must be divided by 65536 since it is given in TDC counts. Therefore, in order to obtain a time measurement in ns the equation 6.1 is applied.

$$t(ns) = [TIME(31:16) + \frac{TIME(15:0)}{65536}] \cdot 100 \quad (6.1)$$

- Temperature data: the temperature data word is a 16 bits word, with the most significant five bits always zero and the remaining eleven bits, TEMP(10:0), representing a signed number. These eleven bits signed number must be processed following the equation 6.2.

$$temp(^{o}C) = 0.12 \cdot TEMP(10:0) - 39.8 \quad (6.2)$$

- Calibration configuration data: these data corresponds to the charge to be programmed in each Beetle chip to generate a calibration pulse corresponding to that charge. The user will introduce the calibration value limits in electrons and the step size also in electrons. The minimum charge will be zero electrons and the maximum 261375 electrons. The minimum step size corresponds to 1025 electrons. All these values are checked by the software. The low level software will send for each calibration acquired data the charge to be acquired in a binary value of 8 bits (0 to 255). Therefore, the equation 6.3 is used for processing these data with the low level software.

$$charge_{binary}(7:0) = \frac{charge_{user}(electrons)}{1025} \quad (6.3)$$

- Trigger Input configuration data: these data will configure the trigger inputs scheme. Therefore, the user will be able to select if the digital input trigger signal will be used (Trig Pulse In), if any of the two photomultiplier input trigger signals (Trig In 1 or Trig In 2) will be used or if the coincidence of the two photomultiplier input trigger signals (Trig In 1 and Trig In 2) will be used. From this information the software generates the required 8 bits data word for the FPGA firmware (0…0 CNIM NE).

- DAC configuration data: these data will configure the four DAC discrimination thresholds. The user will select the value in mV for each threshold (from -2048 mV to 2048 mV in 1mV steps). The low level software will translate these data into the data format required by the FPGA firmware which is a 16 bits word with the address of the corresponding DAC and the value to be programmed (0 0 AD(1:0) DATA(11:0))

These low level objects are in communication with the high level software in order to provide the required data. These required data can be the processed data which have been previously acquired by the hardware and they are used to refresh the GUI plots or to be stored in the corresponding output data file. They also will provide information about the system status to the high level software for generating information for the user. On the other hand, the high level software also sends data to the low level objects in order to generate the required configuration data and the control codes for the FPGA firmware from the user selections at the GUI.

6.3. High level software

The high level software is composed of a GUI for the communication between the user and the system as well as the file generation module for storing the data acquired by the system with a specific format. The GUI is launched when the program is executed. In the figure 6.2 the initial display of the GUI is shown. The GUI can be divided into three different parts: an information console, the current data monitoring screen and the control menus.

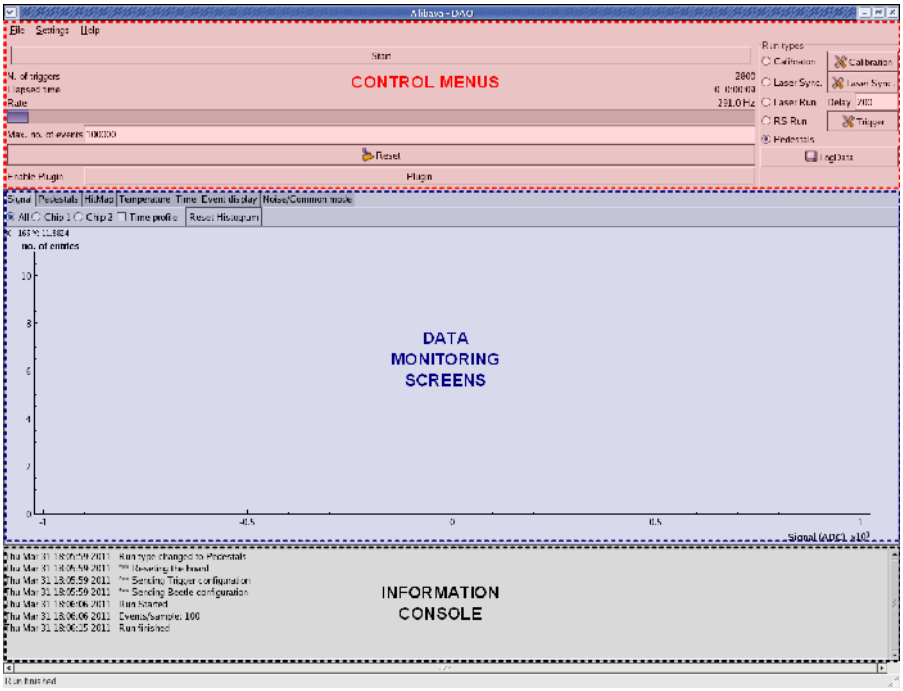

Figure 6.2. *Main windwow of the GUI when it is launched.*

The information console informs to the user, by means of written codes, about the actions taken by the system and the system status. The software operates with a default values for configuring the Beetle chips, the trigger input scheme, the DAC thresholds as well as other software values. When the software is launched, the hardware has been powered on and connected to the PC by USB previously. Then the software resets the system and configures the system with these default values. All these operations are shown in the information console.

By means of the control menus, the system can be controlled and configured fully by the user. Thus, the user can select the run type among the different ones

available (Calibration, Laser Synchronization, Laser Run, Radioactive Source Run and Pedestals). By default the run type selected is the pedestals acquisition since the system must acquire these pedestals of the Beetle signals to be used with the following acquisitions. In the figure 6.3, the Pedestals data monitoring screen is shown in a pedestals acquisition.

The maximum number of data to be acquired for each run type can be introduced in the *Max. no. of events* box of the control menus. Furthermore, in these control menus there are two buttons, one for starting and stopping each acquisition (*Start button*) and another one for resetting the system by software (*Reset button*). On the other hand, there is other information as the number of events acquired (*N. of triggers*), the time elapsed form the beginning of the run (*Elapsed time*) and the acquisition rate (*Rate*).

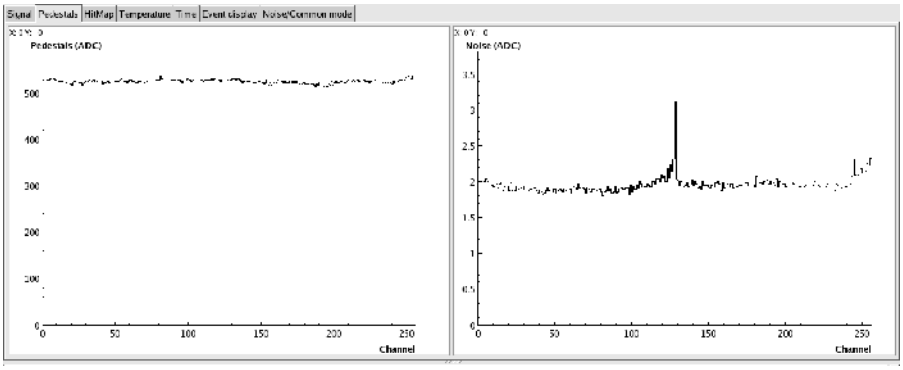

Figure 6.3. *GUI of the software during a Pedestals run.*

In the run types menu of the control menus, several parameters can be configured for each run type by means of the buttons on the right of each run type (*Calibration* button, *Laser Sync*. button and *Trigger* button). In the calibration configuration, the upper and lower range limits of the charge (in electrons) to be programmed in the Beetle chips for calibrating the system can be introduced by the user. By default the system will calibrate from zero to 102500 electrons in 2050 steps. Also the number of samples to be acquired for each charge step can be configured although by default 100 samples by charge step will be acquired. The channel of the Beetle chips to be monitored also can be selected (from channel 1 to channel 256, with channels 1-128 corresponding to one Beetle chip and channels 129-256 corresponding to the other Beetle chip) in the *Settings* menu. By default the channel 60 is monitored. The signal data monitoring screen is shown for a calibration run (with the default values) in the figure 6.4.

In the laser synchronization configuration, the upper and lower limits of the

delay scan can be introduced by the user (in nanoseconds) as well as the delay step (also in nanoseconds). Moreover, the number of samples to be acquired for each delay step can be also configured. For the laser run, there is a box where the delay (in nanoseconds) to be programmed can be introduced by the user. The system will acquire samples with the laser setup according to this delay.

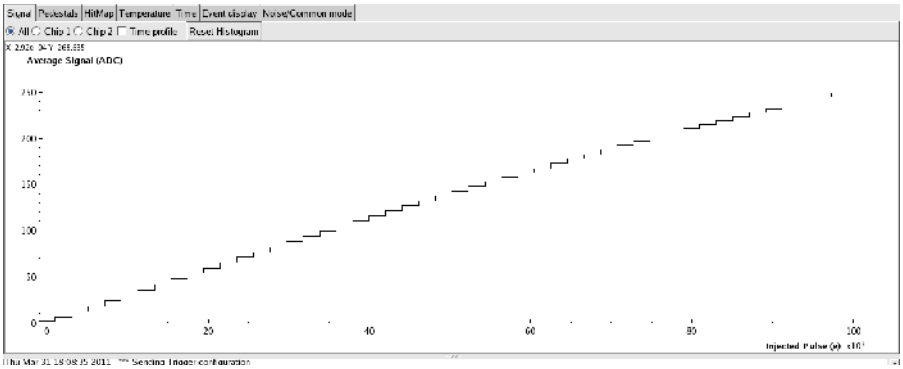

Figure 6.4. *GUI of the software after a Calibration run.*

In the RS run, by means of the Trigger button, the trigger inputs scheme (*i.e.* which combination of trigger inputs are going to be used as trigger input signal) as well as the DAC thresholds can be configured by the user. All these configuration data can also be introduced by means of the *Settings* menu.

Regarding the data monitoring screens, these screens are used to shown different data acquired by the system to the user depending on the run type. The *Signal* screen for the calibration run and the *Pedestals* screen for the pedestals run have been already shown in the figures 6.3 and 6.4 respectively. Some of these screens are only used with specific run types as in the case of the pedestals screen for the pedestals run or the time screen for the RS run. The rest of the screens are used with different run types. The *Signal* screen shows different plots according to the run type. Thus, it shows the calibration data (ADC counts versus injected charge) for a selected channel in the case of a calibration run (figure 6.4), the average signal (ADC counts) versus the delay (nanoseconds) in the case of a laser synchronization run, the number of entries versus the average signal (ADC counts) for a specific delay in the case of the laser run as well as the pedestals level (ADC counts), and the noise level (ADC counts), versus the channel number in the case of the pedestals run (figure 6.3).

In the case of the RS run, two signal screens can be selected. First, the number of entries versus the average signal in ADC counts (figure 6.5). Second, if *Time profile* is selected, the average signal in ADC counts versus the TDC time

(nanoseconds). The data coming from both Beetle chips can be plotted if the *All* radio button is selected, or form individual chips by clicking in *Chip1* button or *Chip2* button.

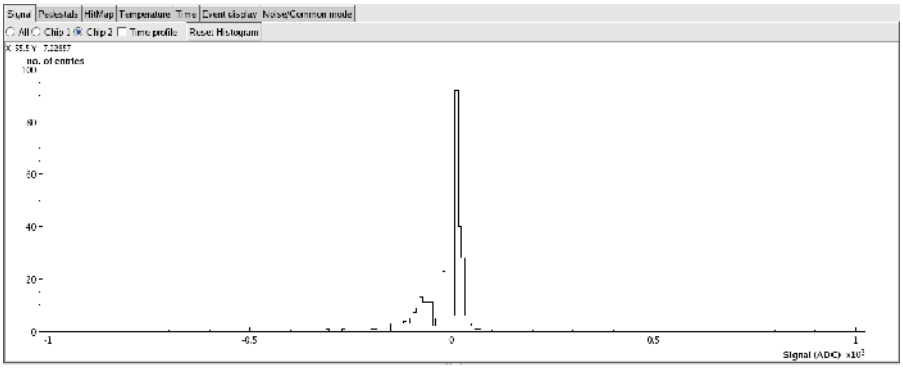

Figure 6.5. *GUI of the software during a RS run with the Signal screen selected.*

The *Pedestals* screen (figure 6.3) shows two plots. On the left side, the the pedestals level (ADC counts) versus the channel number is plotted. On the right side the noise level (ADC counts) versus the channel number is shown. The noise value showed is computed as the RMS (*Root Mean Square*) of the pedestal distribution.

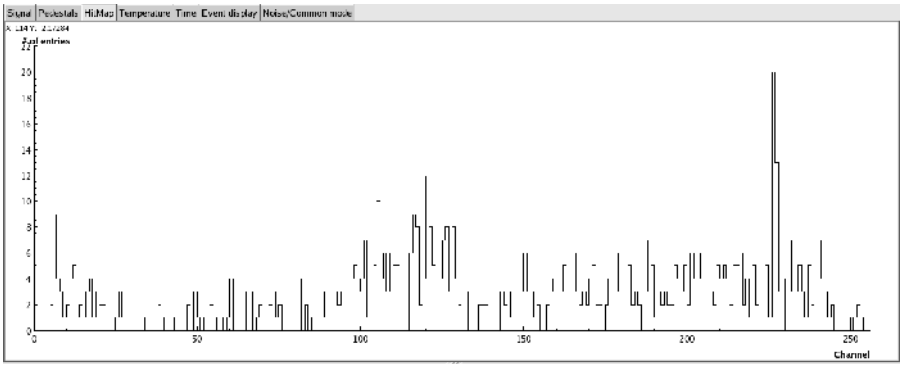

Figure 6.6. *GUI of the software during a RS run with the Hitmap screen selected.*

The *Hitmap* screen is used in the laser synchronization run, the laser run and in the RS run (figure 6.6). It shows the number of entries versus the channel number considering a noise threshold to discriminate real hits from noise. It is useful in order to know which channels have been hit. The *Temperature* screen is used in the pedestals run, the laser run and the RS run. It shows the temperature acquired (in Celsius degrees) at the daughter board for each sample.

The *Time* screen is used just for the RS run (figure 6.7). It shows the TDC measurement (in nanoseconds) for each sample. The *Event display* screen is used with all the run types. In the figure 6.8, this screen is shown for a calibration run. The digitalized Beetle data output (in ADC counts) versus the channel number are plotted in this screen. It is similar to a scope screen but it is not refreshed for each event. The user can choose to see the raw data, without pedestals and common mode correction, or the processed data by clicking on the proper button (*Raw data* or *Signal*) on the histogram.

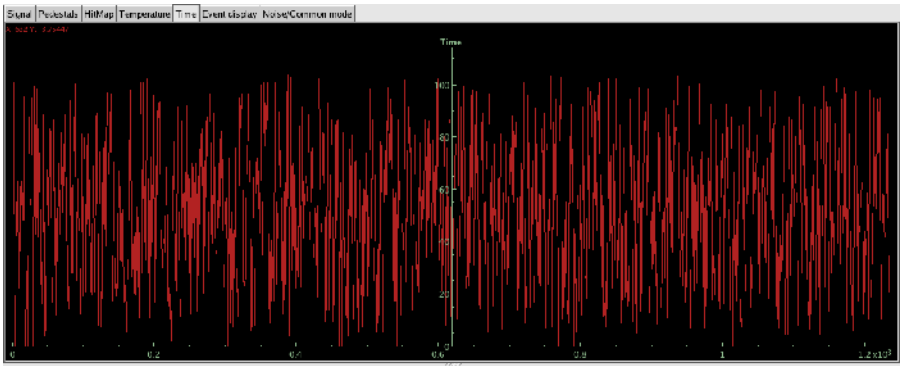

Figure 6.7. *GUI of the software during a RS run with the 'Time' screen selected.*

Finally, the Noise/Common mode screen is used in the laser run and the RS run. In the figure 6.9, this screen is shown for a calibration run. On the left side the average noise (in ADC counts) versus the event number is plotted for each chip. The noise in this plot is computed as the RMS of the channels without signal. On the right side, the common mode (In ADC counts) versus the event number is plotted for each chip.

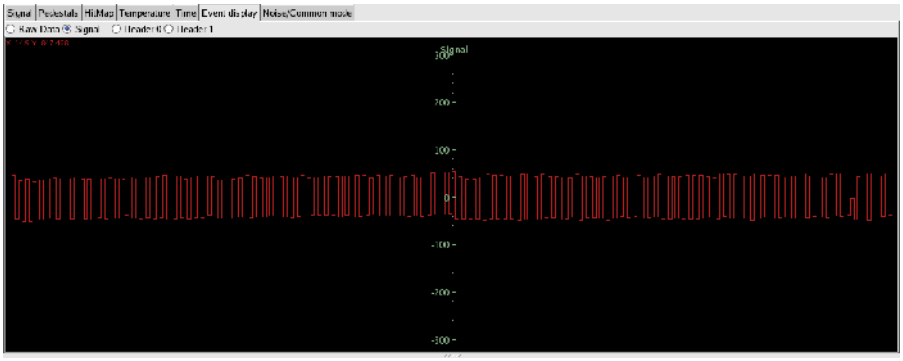

Figure 6.8. *GUI of the software during a RS run with the 'Event display' screen selected.*

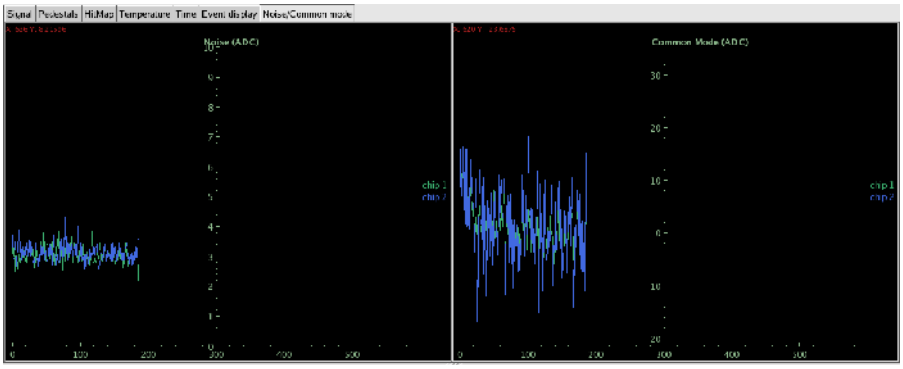

Figure 6.9. *GUI of the software during a RS run with the 'Noise/Common mode' screen selected.*

Regarding the data output file generation, this module is launched by means of the *LogData* button. Thus, the data acquired can be stored for the calibration run, the laser synchronization run, the laser run as well as the RS run. For the pedestals run, it is not necessary to store the data acquired since the software uses these pedestals data for the other run types. However, the pedestals data can be saved on a separate file via the *Save Pedestals* item in the *Settings* menu.

The data output file generated varies for each run type since the data acquired for each run type also is different. However, the format of these files is common with a header informing about the run type and containing configured parameters for that run type. After the header, the file generation module will write in the output data file the data acquired according to the run type. All the data output files are written in binary format. A detailed description of the data format of the output files can be found on [111].

Finally, together with the acquisition software, a number of data analysis algorithms have been developed in the ROOT framework. Taking the data file as input they produce different outputs as plots with different information depending on the macro and the input data file. These plots are useful in order to present and analyze the results from the data acquired as it will be show in the next chapter.

Chapter 7

Development, production and improvement of the system

The development of the system, its production as well as some improvements carried out in the system are treated in this chapter. The development process of the system is summarized in section 7.1. In section 7.2, measurements performed with the system are presented. These measurements were done both with a laser setup and a radioactive source setup. Different types of non-irradiated and irradiated microstrip silicon sensors were used to analyze the system performance. Section 7.3 is devoted to explain the production process of the system, including the quality tests carried out. Finally, the improvements implemented in the system are detailed in section 7.4.

7.1. Development of the system

A prototype of the hardware was developed in order to check the functionality of its design as well as to test and debug the software. The first tests of the hardware part were carried out using custom algorithms designed with the MATLAB [112] environment. These algorithms were very useful for debugging the FPGA firmware of the mother board before the software part of the system was finished. The hardware prototype used for the development tests consisted of a daughter board fully populated, although with no detector connected to the Beetle chips, and a mother board also fully populated. Both the daughter board and the mother board prototypes were almost identical to the final hardware versions since the design

was successfully validated and minimum hardware design changes were required. Some minimal design changes were carried out in the FPGA custom logic blocks and in the embedded processor firmware, for achieving and improving the system functionality.

Firstly, the *Beetle Slow Control block* of the FPGA was designed just for writing the Beetle chips registers without reading back the registers contents. After checking that the Beetle chips could be configured correctly, the *Beetle Slow Control block* of the FPGA was modified in order to read back the Beetle chips configuration registers contents. Thus, the system verifies by itself that the registers are configured correctly.

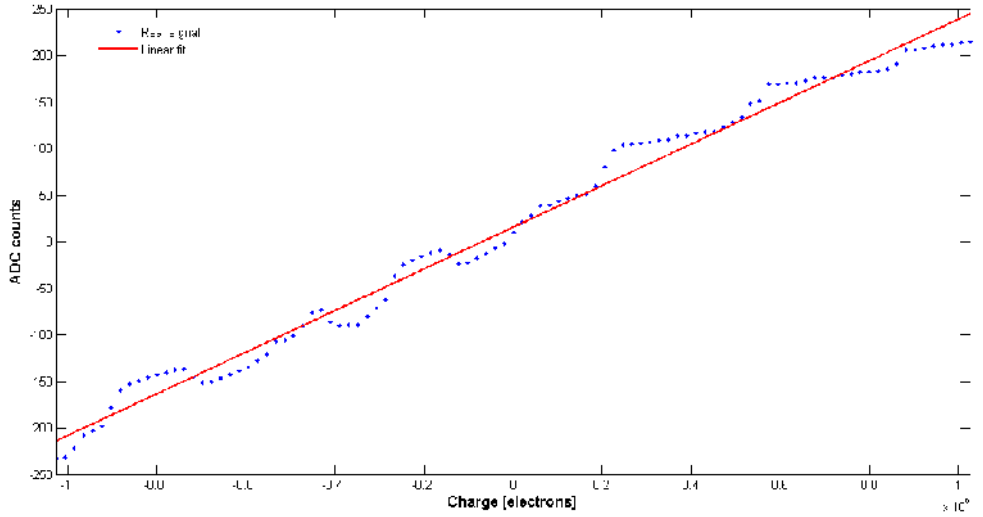

Figure 7.1. *Calibration graph obtained with the motherboard prototype. The averaged signal digitalized (ADC counts) is represented versus the injected charge (electrons). A linear fit for the calibration data (red line) is represented as well.*

Secondly, the synchronization of each ADC of the mother board was adjusted. A calibration between -102500 electrons and 102500 electrons in 2050 electrons steps was carried out. Fifty samples per step were stored. In this calibration, a fixed pedestal value of zero was considered. As a result, a calibration graph was elaborated, which is shown in the figure 7.1. The averaged signal digitalized in ADC counts is represented versus the injected charge in electrons. As seen in the graph, the obtained values follow a linear fit as it was expected and they are in expected range. However, the measured values represented form some waves following the linear fit.

The main reason of these waves was understood as the synchronization of the sampling clock of each ADC was not optimum, sampling in the first 5 ns of each multiplexed channel. Therefore, this fact produced these variations in the digitalized values since the sampling point for each digitalized value was inadequate (*i.e.* the greater variations in the signal for each channel take place in the first 5 ns and in the last 5 ns). In order to fix this issue, the synchronization of the sampling clock signal was optimized in order to sample just in the middle of each multiplexed analogue channel (with a resolution of ±3.125 ns), as it is shown in the figure 7.2. This modification was carried out in each *ADC Control block* in the FPGA by delaying the sampling clock in 5 ns steps with a flip-flop register clocked at 200 MHz.

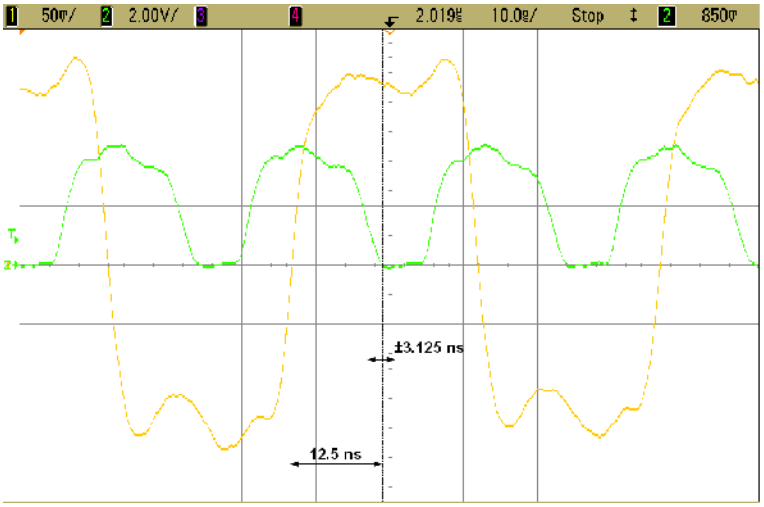

Figure 7.2. *Beetle output analogue signal (yellow signal, channel 1) in a calibration and the corresponding ADC sampling clock signal of 40 MHz (green signal, channel 2) with optimum synchronization.*

Thirdly, both *Pedestals Acquisition* and *Pedestals Reading* states were implemented in the FPGA in order to include the pedestals of the Beetle analogue output signals in all the acquisitions. This was important for improving the calibration values since the pedestals of the Beetle signals were not equal to zero as it was supposed (*i.e.* the signal has a level different to zero when no charge is acquired).

Finally, the gain of the system was increased. It was assumed that the unity gain in the Beetle output analogue signal path (which corresponds to a signal of *38 mV/22000 electrons* under nominal bias parameters and 25 ºC, as stated in equation 4.1) could not be enough in order to take acquisitions with heavily irradiated

silicon sensors and a $^{90}Sr/^{90}Y$ β source (*i.e.* typical signals of a *mip*, or about 22000 electrons). Therefore, the gain was doubled to have more resolution for measurements with a radioactive source but enough dynamic range for measurements with a laser (*i.e.* with much higher than a *mip*).

The gain at the mother board was doubled by halving the full scale voltage of the ADCs from ±1024 mV to ±512 mV. This was carried out by reducing VREFIN from 2048 mV to 1024 mV (adding a resistor of 10 kΩ to form a resistive divider). This option minimizes the noise generation with respect to other options (*i.e.* doubling the gain of the mother board single-ended to differential amplifiers or doubling the gain of the daughter board differential line drivers). The resolution of each ADC would change from 2 mV/ADC to 1 mV/ADC.

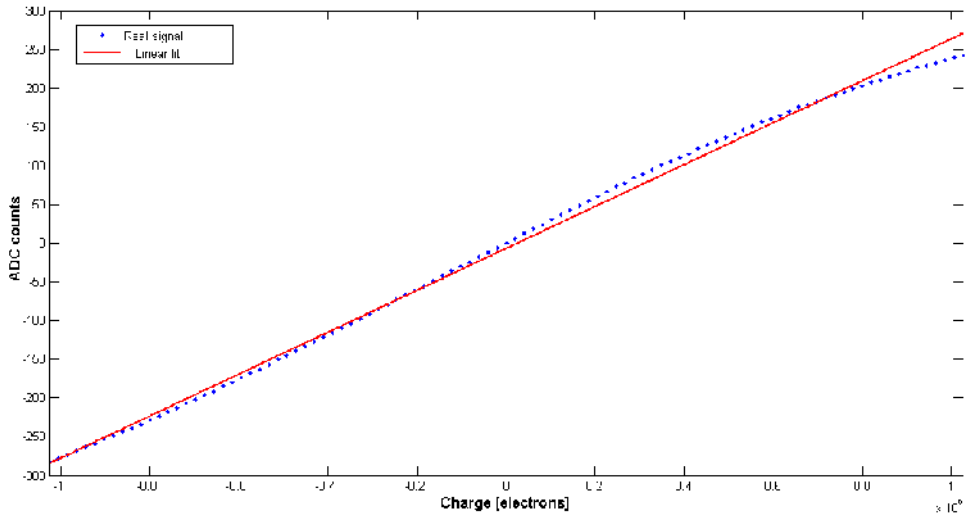

Figure 7.3. *Calibration graph obtained with the motherboard prototype optimized. The averaged signal digitalized (ADC counts) is represented versus the injected charge (electrons). A linear fit for the calibration data (red line) is represented as well.*

Taking into account these changes, a new calibration acquisition was carried out. As a result, a new calibration graph was elaborated, which is shown in the figure 7.3. In this graph the averaged signal digitalized in ADC counts is represented versus the injected charge in electrons. As seen in the figure 7.3, the oscillations in the calibration values disappeared and these values fit well with a linear behaviour. The data represented in this graph agree with the gain specification for the Beetle chip front-end (38 mV/mip under nominal bias parameters and 25 ºC) since the signal corresponding to ± 22000 electrons is about ±76 ADC counts with a gain of two.

7.2. Measurements with sensors

After performing the development tests with the prototype of the system, an initial batch of five mother boards was produced and populated. Fifty daughter boards were also produced and some of them populated. Three of these daughter boards were assembled with three detector boards, three test boxes and the corresponding pitch adaptors in order to connect different types of detectors. These new mother boards and the assembled daughter boards were used for performing measurements with different types non-irradiated and irradiated microstrip sensors. The results of these measurements are described in the following sections.

7.2.1. Processing of the acquired data

It is worth explaining how the data is processed when a laser acquisition or a radioactive source acquisition is carried out for a better understanding of the results presented in the following sections.

The software generates an output data file where the data acquired on the different run types (calibration, laser synchronization, pedestals acquisition, laser acquisition or radioactive source acquisition) are stored for further processing. These files contain a header with the information about the run type and the parameters configured for that run type as well as the data acquired in binary format. After the header, the type of data stored will vary depending on the run type, having in all the run types at least the data digitalized from the Beetle chips corresponding to each input channel (1 to 256) for each event. Other types of data stored for each event according to the run type could be the charge injected in electrons (calibration run), the TDC measurement in ns (radioactive source run), the delay implemented in ns (laser synchronization) or the temperature in ºC (laser and radioactive source acquisition). The pedestals are computed on-line. The data acquired in a pedestals run, *i.e.* the average of the acquired data in ADC counts for each channel (pedestal level) and the RMS of the acquired data in ADC counts for each channel (noise), are stored in a text file. These data can also be stored when the data files corresponding to other run types are generated, as binary data after the header if pedestals data have been acquired with the software previously. A detailed description of the format of these output files can be found on [111].

A set of macros have been developed for the ROOT framework [109] to perform the required processing in order to show the data in an adequate format. They take the file where the data (acquired in a laser synchronization run, laser run or radioactive source run) has been stored in a raw format (ADC counts versus

input channel for each event) and they use also additional files (from a calibration run and a pedestals run) to process the data.

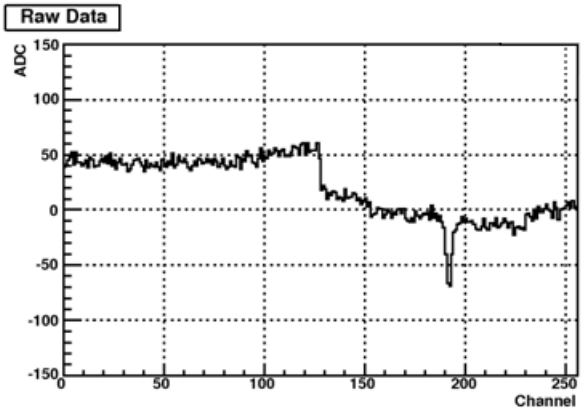

Figure 7.4. *Digitalization of the Beetle analogue multiplexed output signal for an event. ADC counts versus input channel number.*

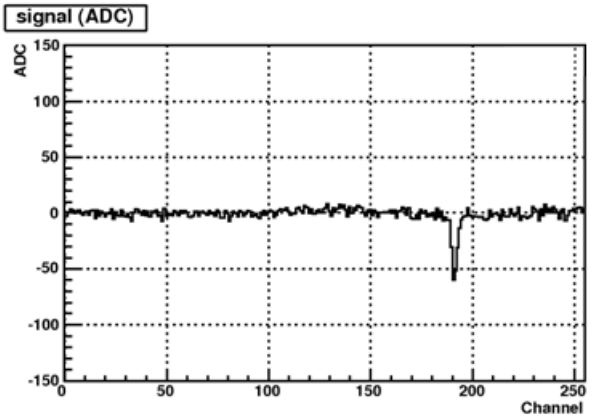

Figure 7.5. *Digitalization of the Beetle analogue multiplexed output signal for an event, with the pedestal value substracted for each channel. ADC counts versus input channel number.*

For each event, a trigger generated in the laser acquisition (or laser synchronization) or a trigger processed in the radioactive source acquisition, the analogue multiplexed output data of each Beetle chip is digitalized. Therefore, a raw data plot can be obtained for each event (figure 7.4). These raw data are corrected by subtracting the pedestal value and by performing a common mode correction. The result is represented in figure 7.5. From these corrected data, the signal is computed as the sum of strips in a cluster. Clusters are built around strips

with high SNR (*signal to noise ratio*) which are not already in a cluster, called seeds. A strip would become a seed if its SNR is higher than 6. Then the adjacent channels are added while their SNR is higher than 2.5. Once the signal is computed, different representations of the acquired data can be obtained from a given number of events. At this point the calibration data (relationship between ADC counts and electrons for each channel) can be used in order to obtain the data represented in electrons. The representations which can be obtained depend on the acquisition type and can be the signal spectrum (laser acquisition and radioactive source acquisition), the signal spectrum with a time cut (radioactive source acquisition) and the analogue pulse shape reconstruction of the Beetle front-end (laser synchronization and radioactive source acquisition).

7.2.2. Measurements with non-irradiated detectors

Two non-irradiated microstrip silicon sensors of 300 μm of thickness and 128 channels were bonded to two daughter boards for determining the performance of the system. One detector was a *n*-type detector (*p*+-*n*) whereas the second one was a *p*-type detector (*n*+-*p*). The pitch of the detectors was 80 μm and the length of the strips 1 cm. In each daughter board used, one Beetle chip was connected to the detector (channels 129-256) and the other chip was not connected (channels 1-128). Two types of measurements were performed: measurements with a laser system and measurements with a β source. The pedestals and the calibration data were acquired for each daughter board before. All these measurements were carried out at 20 ºC with the detectors biased at full depletion voltage (-100 V for the *p*-type detector and 200 V for the *n*-type detector).

For the calibration measurements, the acquired data corresponds to internal injected charge by each Beetle chip. The amount of injected charge is programmed in the Beetle chips via I2C. In particular, charges from 0 electrons to 102500 electrons were injected in 1025 electrons steps. For each step 100 samples are acquired. The sign of the charge alternates with the channel number. From these data a calibration curve, like the graph shown in figure 7.3, can be generated for each channel. Then, the gain for each channel can be derived. The ADC counts per electron (*i.e.* the gain) versus the channel number are shown in figure 7.6 for the daughter board with the *p*-type detector connected to channels 129-256. The different conversion factors are due to the capacitive loading of the second 128 channels by the strips. The input capacitance in the input channels of the Beetle chip with the detector connected is higher and it has influence on the measurements. The plot shows that the conversion factor does not change in a significant way for the channels corresponding to the detector, which was the expected behaviour. In the figure 7.7, the same plot is represented for the daughter

board with the *n*-type detector connected to channels 129-256.

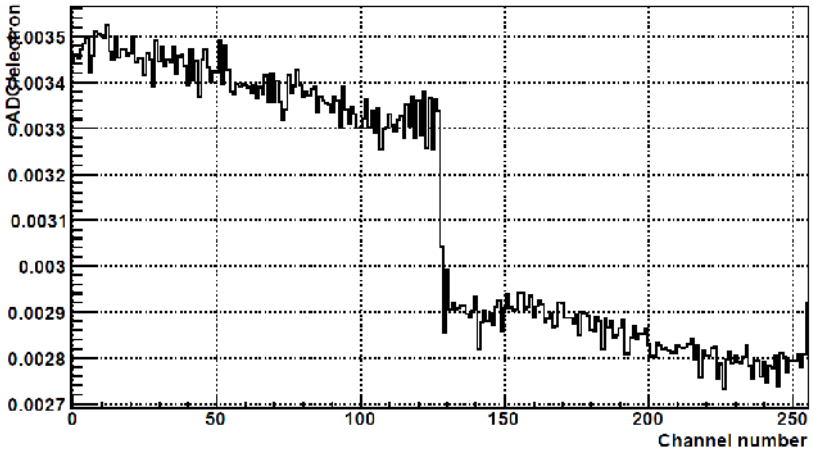

Figure 7.6. *ADC counts per electron (gain) versus channel number for the non-irradiated p-type detector connected to one Beetle chip (channels 129-256).*

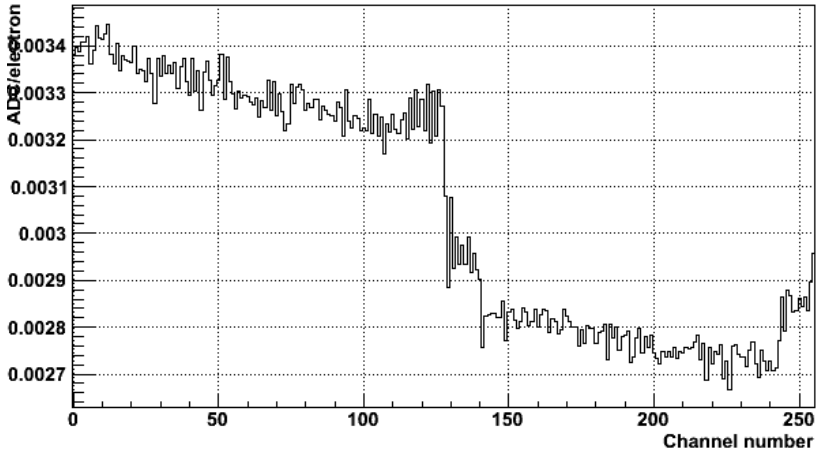

Figure 7.7. *ADC counts per electron (gain) versus channel number for the non-irradiated n-type detector connected to one Beetle chip (channels 129-256).*

Measurements using a laser setup were carried out for both detector types. A near infrared laser with wavelength of 1060 nm and a photon energy of 1.17 eV was used. A laser scan was carried out with a programmed delay range of 100 ns (from a delay of 1040 ns to a delay of 1140 ns) in 1 ns steps. A hundred of samples per step were acquired. This delay is found experimentally once the laser beam has been focused on the detector by trying different delay ranges with larger delay steps. Once the signal is found, both delay range and the delay step can be

constrained in order to reconstruct the Beetle front-end analogue pulse shape generated by the detector signal from the laser light pulse. In the figure 7.8 the collected charge in electrons versus the delay in ns is represented for the *n*-type detector whereas the figure 7.9 corresponds to the *p*-type detector. It can be seen that the analogue Beetle front-end pulse has been reconstructed with a peaking time about 25 ns and the subsequently undershoot. For the *n*-type detector the pulse is positive, i.e. holes are collected in the detector, while for the *p*-type detector the pulse is negative.

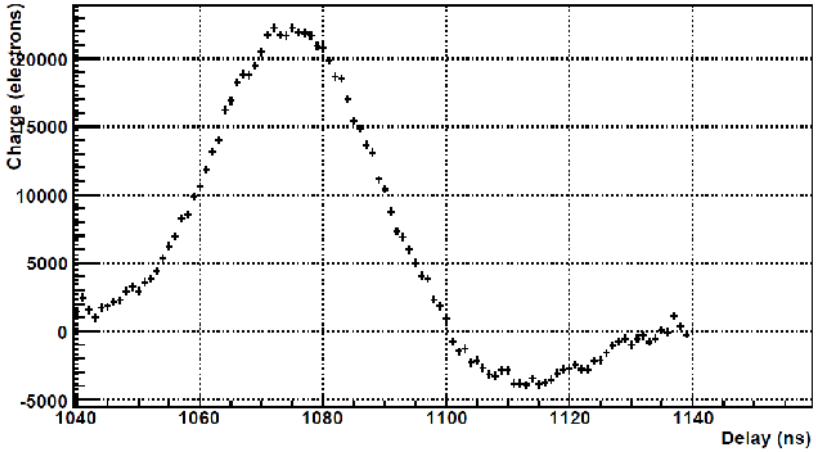

Figure 7.8. *Laser scan for the non-irradiated n-type detector where the Beetle front-end pulse shape is reconstructed.*

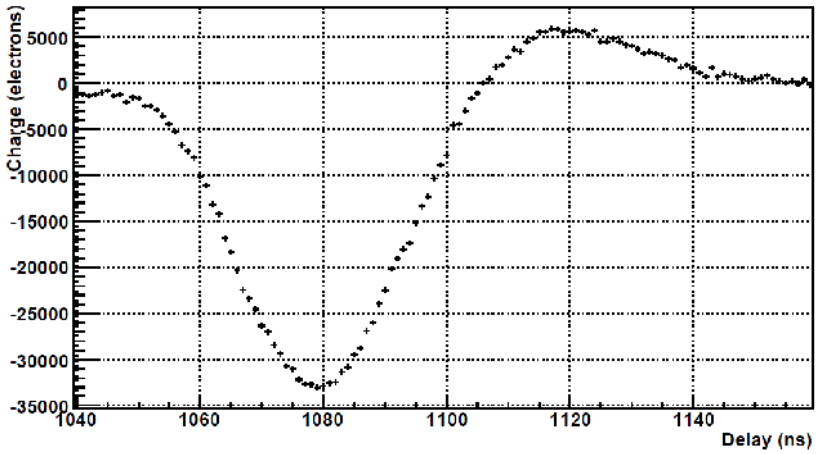

Figure 7.9. *Laser scan for the non-irradiated p-type detector where the Beetle front-end pulse shape is reconstructed.*

Finally, measurements with a low activity $^{90}Sr/^{90}Y$ β source were acquired with both detectors. As a trigger input for this measurement, the output signal from one photomultiplier connected to a scintillator, placed under the detector and the radioactive source, was used. This signal was connected to the motherboard TRIG IN1 input and discriminated with a -40 mV threshold level. A number of 20000 samples were acquired with each detector.

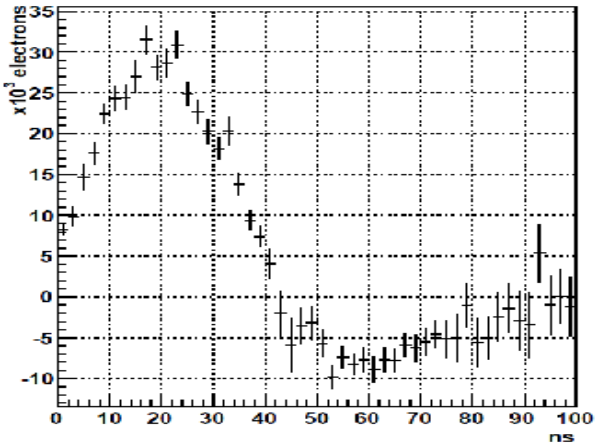

Figure 7.10. *Pulse reconstruction of the Beetle front-end pulse shape for a n-type detector from measurements with a β source.*

In the figure 7.10 the reconstruction of the Beetle analogue front-end pulse shape is shown for the *n*-type detector. The averaged collected charge in electrons is plotted versus the TDC measurement in ns. A peaking time about 25 ns can be noticed. The pulse is positive for the *n*-type detector, *i.e.* holes are collected in the detector.

The spectrum signal, as the number of events versus the charge in absolute value, with a time cut between 12 ns and 22 ns is shown for the *p*-type detector in the figure 7.11 and in the figure 7.12 for the *n*-type detector. This time cut includes events with TDC measurements between 12 and 22 ns. These events correspond to the peak of the pulse shape reconstructed (figure 7.10 for the *n*-type detector). It can be seen in both figures that the spectrum is consistent with a Landau distribution convoluted with a Gauss distribution, with a long tail for greater collected charges. This was the result expected for this type of measurement since it would be directly related to the energy loss in the silicon detector with a thickness of 300 μm. The peak of this distribution (26940 electrons for the *p*-type detector and 24810 electrons for the *n*-type detector) corresponds to the most probable value charge for a mip (*minimum ionizing particle*).

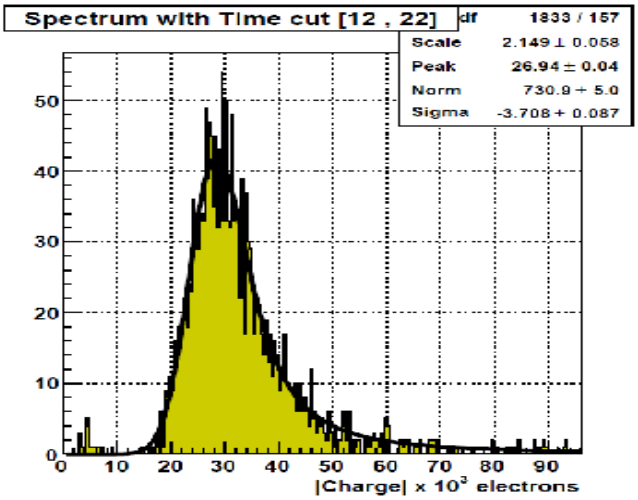

Figure 7.11. *Spectrum of the signal acquired with the p-type detector and a β source with a time cut (12-22 ns).*

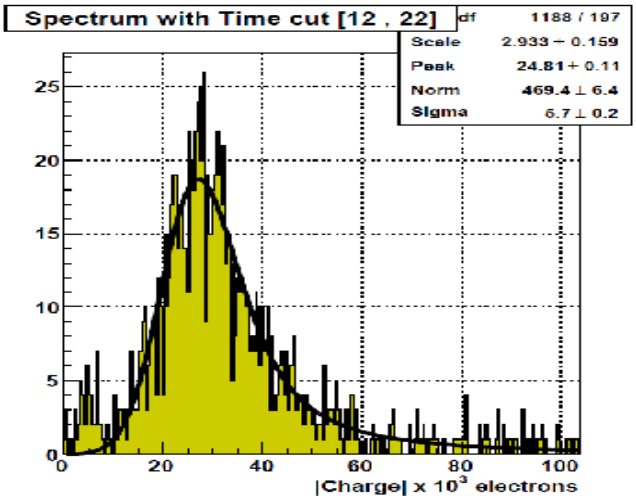

Figure 7.12. *Spectrum of the signal acquired with the n-type detector and a β source with a time cut (12-22 ns).*

The noise corresponding to the daughter board with the *p*-type detector is shown in figure 7.13. The noise value in electrons, as the rms (*root mean square*) value of the pedestal distribution obtained for this daughter board, versus the channel number is plotted in this figure. The channels connected to the detector have a noise level about 690 electrons while the noise level for the channels not connected

to the detector is about 530 electrons. The noise level of the channels connected to the detector corresponds to the sum of the detector noise, the Beetle chip noise and the electronics noise. Therefore, this noise level is higher than the one corresponding to the channels not connected to the detector (*i.e.* just the Beetle chip noise and the electronics noise).

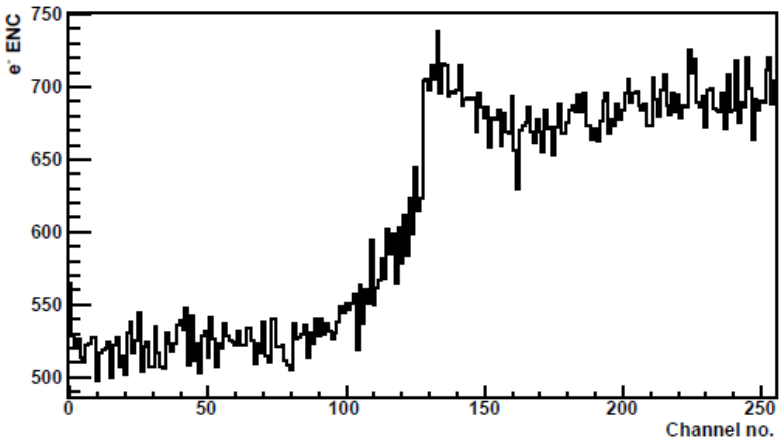

Figure 7.13. *Noise level in electrons versus the channel number for the non-irradiated p-type detector connected to one Beetle chip (channels 129-256).*

Taking into account the peak value of the distribution of the figure 7.11 (26940 electrons) divided by the noise level of the channels connected to the detector (690 electrons from figure 7.13) a SNR of about 38 is obtained. Similar results are obtained for the *n*-type detector.

7.2.3. Measurements with irradiated detectors

Measurements were carried at the IFIC with irradiated *p*-type (*n*+-*p*) microstrip silicon detectors of 320 μm of thickness and 128 channels were performed using both the $^{90}Sr/^{90}Y$ β source and the infrared laser [113]. The same type of measurements with irradiated *n*+-*n* microstrip silicon detectors of 300 μm of thickness and 128 channels were also performed [114]. SNR values higher than 30 were obtained for these measurements with irradiated detectors.

The measurements were carried out at -30 ºC to reduce the noise of the irradiated detectors. The daughter board with a detector to measure bonded to one Beetle chip (channels 129-256) was placed inside a freezer. Firstly, calibration measurements were performed with a non-irradiated detector at 20 ºC and at -30 ºC. The detector was biased at full depletion voltage. Charges from 0 electrons to 102500 electrons

were injected in 1025 electrons steps. For each step 100 samples are acquired. The sign of the charge alternates with the channel number. The calibration data acquired at 20 ºC (ADC counts per electron versus the channel number) are shown in figure 7.14 while the calibration data acquired at -30 ºC are shown in figure 7.15.

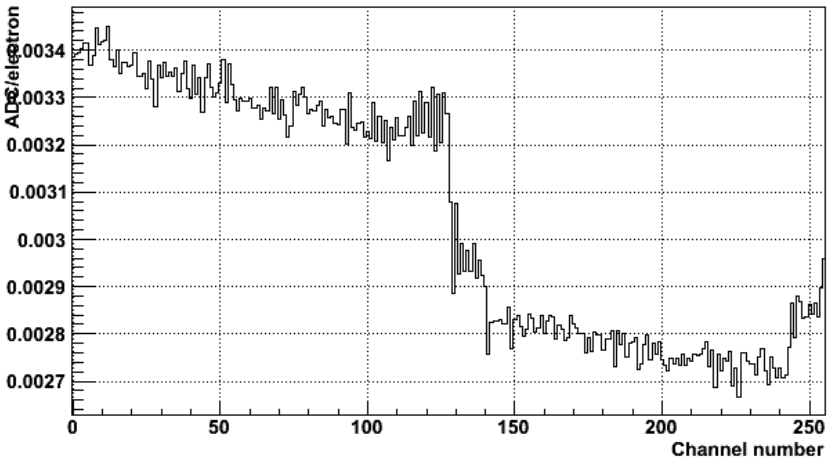

Figure 7.14. *ADC counts per electron (gain) versus channel number for a non-irradiated p-type detector connected to one Beetle chip (channels 129-256). Data acquired at 20 ºC.*

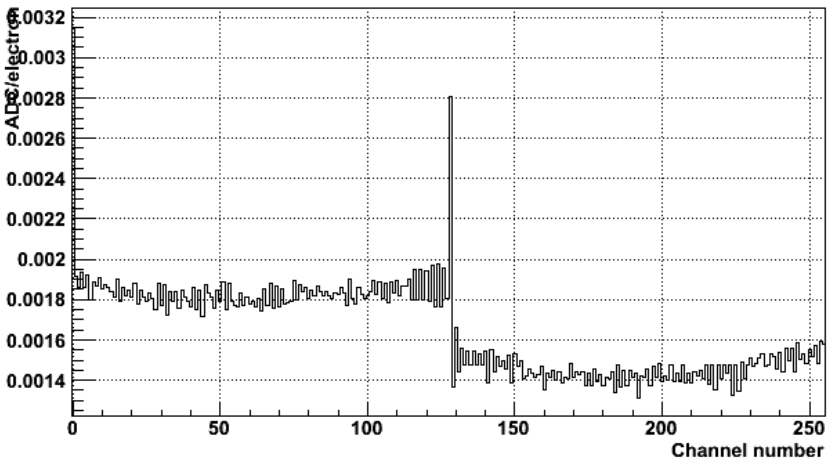

Figure 7.15. *ADC counts per electron (gain) versus channel number for a non-irradiated p-type detector connected to one Beetle chip (channels 129-256). Data acquired at -30 ºC.*

The data shown in figure 7.14 are very similar to those data shown in figure 7.6 ans in figure 7.7, as expected. However, the gain, for both the channels connected to the detector and the channels not connected, is very different for the data taken

at -30 ºC (figure 7.15). Such a gain change was not expected since the gain of the Beetle front-end at -30 ºC should be about 1.4 times higher than the gain at 20 ºC [44]. However, the gain of the figure 7.15 for all the channels is about 1.85 times lower than the gain of the figure 7.14. Therefore, it was decided to use the calibration data obtained with the non-irradiated detectors fully depleted at 20 ºC for the measurements with the irradiated detectors of the same type. A gain correction factor, R_{cal}, was calculated to relate the data obtained at -30ºC with the data obtained at 20 ºC,

$$R_{cal} = \frac{Q_{20ºC}}{Q_{-30ºC}} \quad (7.1)$$

where $Q_{20ºC}$ is the most probable charge obtained from a β source measurement with the non-irradiated detector at 20 ºC (for instance the peak value of figure 7.11) and $Q_{-30ºC}$ is the most probable charge obtained from a β source measurement with the same detector at -30 ºC. Thus, the calibration data obtained at 20ºC could be used for the measurements carried out with at -30ºC. After investigating the reason of this unexpected gain change of the calibration data, the system was improved in order to be able to use the calibration data obtained at low temperatures, as it is explained in section 7.4.1.

Daughter board production batches		
Batch number	**Date**	**Number of boards**
1	10/2007	50
2	12/2010	76
Mother board production batches		
Batch number	**Date**	**Number of boards**
1	12/2007	5
2	11/2008	22
3	11/2009	20
4	11/2010	24

Table 7.1. *Summary of the daughter board and mother board production batches.*

7.3. Production of the system

After measuring with different types of microstrip silicon sensors and checking that the performance of the system was good enough the production of the system was launched. The initial batch of 5 mother boards and 50 daughter boards was intended for internal use of the ALIBAVA collaboration. However, several

members of the RD50 collaboration showed their interest in acquiring the system. A new production batch of 22 mother boards was launched in November of 2008 to cope with this demand. Some of these systems were also distributed to research groups not involved in the RD50 collaboration. Two new batches of 20 and 24 mother boards as well as a new batch of 76 daughter boards were produced in November of 2009 and December of 2010 due to the further interest in acquiring the system of other research groups.

7.3.1. Production process and quality tests

The daughter board production consists of several steps. First, the PCB (*Printed Circuit Board*) is manufactured by an external company from the *Gerber* files where the board design is specified. Second, the board is populated by another external company. This company is also in charge of the purchase of the components from the component list. Finally, the Beetle chips and the fan-ins are glued and bonded in the daughter board (figure 7.16). This process has been carried out in firstly at the University of Liverpool and afterwards at the IFIC (*Instituto de Física Corpuscular*). Some daughter boards have been also bonded by an external company.

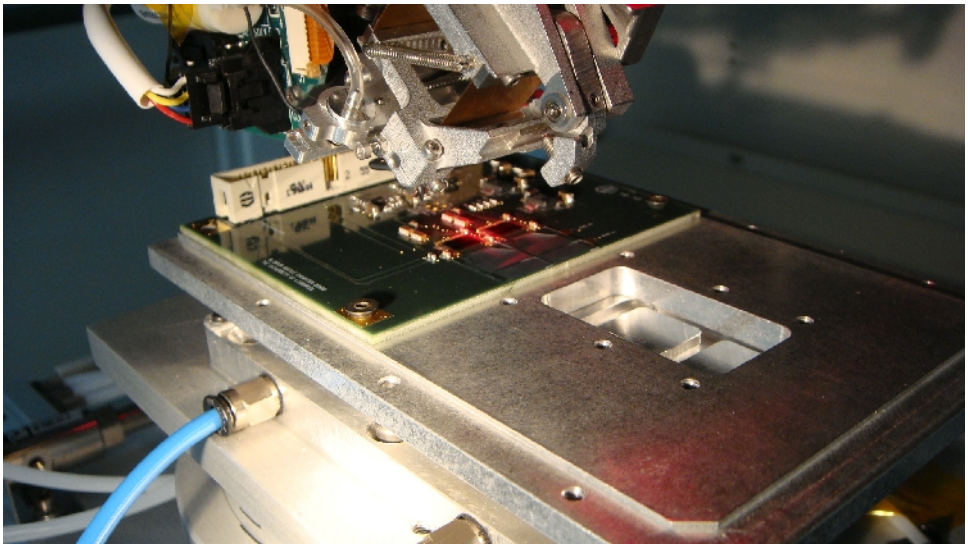

Figure 7.16. *Daughter board bonding process.*

After the daughter board has been fully populated, some quality tests are performed to verify the operation of each daughter board. Firstly, some critical points of the board (supplies and grounds) are checked for possible short circuits

with a multimeter. This is done before and after bonding the Beetle chips. Secondly, the daughter board is powered on (by connecting it to a mother board) and all the power supply levels of the board are checked with a multimeter. Finally, an operation test is performed with the daughter board connected to a fully operational mother board. The mother board is also connected to the software. The chips are configured and a pedestals run is carried out. A calibration run is also performed. The analogue Beetle signal is checked with the software and the scope for the pedestals and the calibration run.

If all the quality tests are passed, the daughter board is marked with a label identifying the batch and the board number. Finally, the board is screwed to the aluminum test box. The input ground of the daughter board is connected to the test box. The box is fabricated by an external company.

The mother board production has been also divided in two processes. Like with the daughter board, the PCB is manufactured by a company from the *Gerber* files. Then, the board is populated by another company which also purchases the components from the component list.

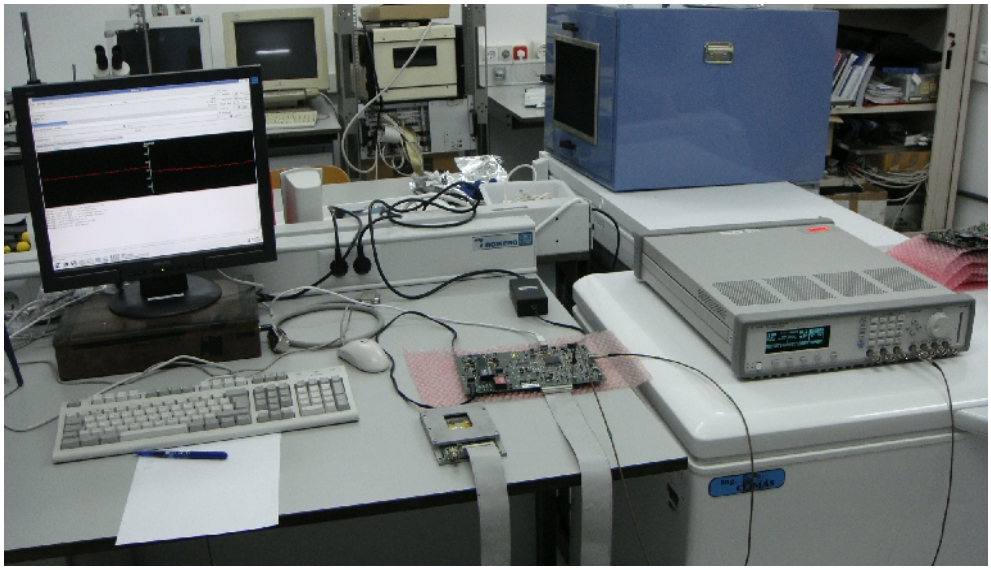

Figure 7.17. *Laboratory set-up for the operation quality test of the mother board and the daughter board. The mother board is connected to a fully operational daughter board and also to the software by USB. A pulse generator is used for emulating the external trigger. A scope is also used (not shown in the picture) to check different signals.*

Some quality tests are performed in order to assure the correct functionality of

each mother board. Firstly, some critical points of the board (supplies and grounds) are checked for possible short circuits with a multimeter. Secondly, the mother board is powered on without being connected to the daughter board and all the power supply levels of the board are checked with a multimeter. Thirdly, the on-board PROM memory is programmed with the FPGA firmware by *Boundary-Scan* (JTAG) through a dedicated connector by means of the *Xilinx USB Platform Cable* [115]. The rotary switches of the mother board are positioned to digitalize both positive and negative analogue input signals coming from the Beetle chips. Finally, an operation test is carried out with the mother board connected to a fully functional daughter board with no detectors connected. The mother board is also connected to the software (figure 7.17). All the acquisition types are tested. The chips are configured and a pedestals run is carried out. Then, a calibration run is performed. The analogue Beetle signal is checked with the software and the scope for the pedestals and the calibration run. Afterwards, the laser synchronization and the laser run are tested. The TRIG OUT signal levels and its delay regarding the Beetle analogue signal is checked with a scope. Finally, the radioactive source run is tested. A pulse generator is used for emulating the input trigger pulses with different rates up to 100 kHz. A scope is used to check the analogue multiplexed signal from the Beetle chips.

If all the quality tests are passed the mother board is marked with a label identifying the batch and the board number. Finally, the mother board is screwed to an aluminum box. The input ground of the mother board is connected to the box. The mother board box is fabricated at the IFIC mechanical workshop. The box has been designed specifically for this board.

7.3.2. Distribution and measurements performed with the system.

A number of systems have been distributed among different research groups. In total, 34 systems has been distributed. Moreover, 35 extra daughter boards and 35 extra fan-ins packs. The institutions that have acquired a system are summarized in table 7.2.

With each system the following items are included (figure 7.18):

- 1 mother board with box.
- 1 daughter board with fanins glued as well as a test box.
- 2 detector boards.
- 6 extra fanins (*fan_det* type).

- 1 power supply (ac-dc desktop voltage source).
- 1 USB cable.
- 1 flat cable.
- 2 HV Lemo connectors for the detector power cable.
- Software copy and documentation for using the system.

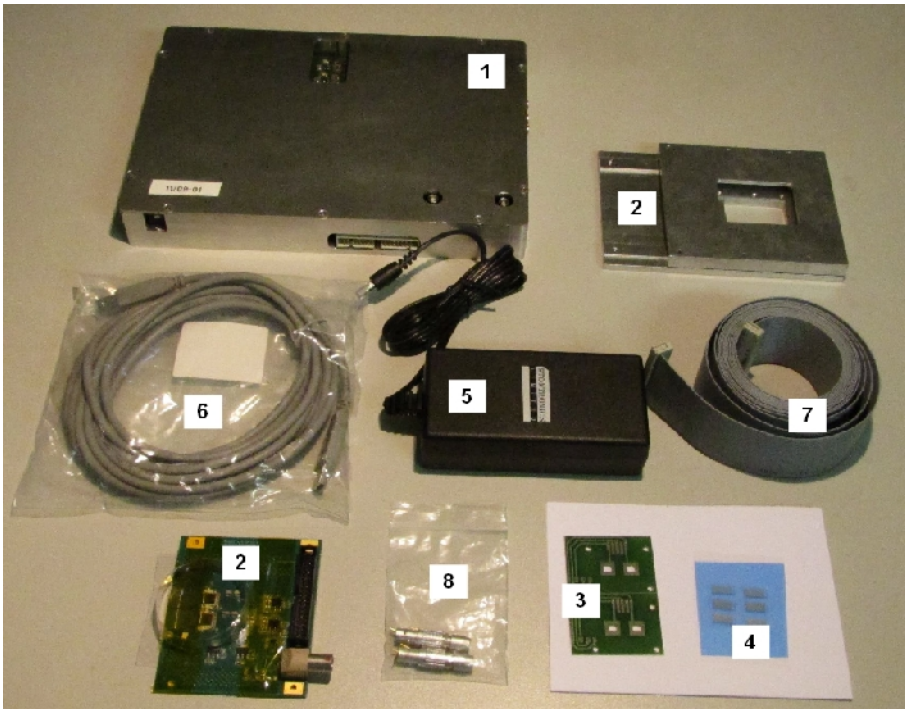

Figure 7.18. *Items included with the system.*

The system has been mainly used to perform measurements with microstrip silicon sensors. For instance, charge collection measurements with *p*+-*n* and *n*+-*p* microstrip silicon sensors irradiated with protons and neutrons (up to $8 \cdot 10^{15}$ 1 MeV equivalent neutron/cm^2) have been carried out by different research groups of RD50 [116-119]. These detectors were 300 μm thick, they have a strip pitch of 80 μm and a strips length of 1 cm. These detectors were coupled to the Beetle chips in *ac* mode. The same type of measurements have been performed with *dc* coupled *n*+-*p* and *n*+-*n* microstrip silicon sensors irradiated with protons (up to 10^{16} 1 MeV equivalent neutron/cm^2) [117, 120]. These detectors had a thickness of 75 and 150 μm, a strip pitch of 80 μm and a strip length of 7 mm.

Other type of sensors have been also connected to the system. For example, charge collection measurements using both an infrared laser and a β source have been performed with *p*-type double sided 3D silicon detectors [121, 122]. These type of detectors had a thickness of 285 μm and they were coupled in *dc* mode. They were irradiated with protons with doses up to $2 \cdot 10^{16}$ 1 MeV equivalent neutron/cm^2. Finally, measurements with both an infrared laser and a β source have been carried out with a *p*+-*n* 2D microstrip silicon detector coupled in *ac* mode [123]. The detector had a thickness of 300 μm.

Institution	Country
CERN	Switzerland
University of Freiburg	Germany
University of Florence	Italy
University of Glasgow	United Kingdom
Charles University Prague	Czech Republic
University of Ljubljana	Slovenia
University of Rochester	United Kingdom
University of Lancaster	United Kingdom
Max-Planck-Institut fuer Physik, Munich	Germany
Instituto de Física de Cantabria	Spain
University of Karlsruhe	Germany
University of Sheffield	United Kingdom
Université catholique de Louvain	Belgium
Universidad de Santiago de Compostela	Spain
University of Heidelberg	Germany
University of Cambridge	United Kingdom
Brookhaven National Laboratory	United States
École polytechnique fédérale de Lausanne	Switzerland
Deutsches Elektronen-Synchrotron	Germany
Czech Technical University in Prague	Czech Republic
Institute of Physics, Academy of Sciences of the Czech Republic	Czech Republic
University of Manchester	United Kingdom

Table 7.2. *Institutions which have acquired a system*

7.4. Improvement of the system

Some improvements have been implemented on the system. These improvements are based on the experience of the users which have taken

measurements with the system.

7.4.1. Daughter board analogue buffers

A problem was detected when operating the daughter board at low temperature. In particular, the calibration data obtained at -30 ºC did not correspond with the data expected (see section 7.2.3). Thus, the gain (ADC/electrons) for the calibration data obtained at -30 ºC was 1.85 times lower than the gain for the calibration data acquired at room temperature. However, the gain for the calibration data taken at -30 ºC should be about 1.4 times higher than the gain for the calibration data taken at 20 ºC.

It was found at the University of Liverpool that this was due to the limited common-mode input range of the daughter board analogue buffers [50]. Originally, the power supply level for these buffers was a single 3 V. This limits the common-mode input range from 0.3 V to 1 V. The common-mode of the Beetle analogue output signal is lower than 1 V at room temperature. However, this Beetle common-mode level increases with lower temperatures, forcing the analogue buffer to operate in a non-linear regime (slewing the signal edges for instance). Therefore, the power supply level for these analogue buffers has been changed to 4.5 V. This increases upper limit of the common-mode input range from 1 V to 2.5 V. This change has been implemented by replacing the LP2985 3V linear regulator [54] by the LM1117-4V5 linear regulator [53]. The second production batch of daughter boards (table 7.1) are populated with this new regulator.

Calibration measurements were taken at room temperature (20 ºC) and at -20 ºC with the new daughter board in order to verify a correct operation of the calibration circuit at low temperatures. The Beetle chips of the daughter board were not connected to any detector. Charges from 0 electrons to 102500 electrons were injected in 1025 electrons steps. For each step 100 samples are acquired. The sign of the charge alternates with the channel number.

The calibration data acquired at 20 ºC (ADC counts per electron versus the channel number) are shown in figure 7.19 while the calibration data acquired at -20 ºC are shown in figure 7.20. It can be seen that the conversion factors of both Beetle chips (about 0.00385 ADC/electron for channels 1-128 and about 0.0037 ADC/electron for channels 129-256) for the calibration data obtained at -20 ºC are about 1.3 times higher than the conversion factors of both Beetle chips (about 0.00295 ADC/electron for channels 1-128 and about 0.0028 ADC/electron for channels 129-256) for the calibration data obtained at 20 ºC. This result corresponds to the gain increase of the Beetle front-end at -20ºC with regard to

20ºC [44].

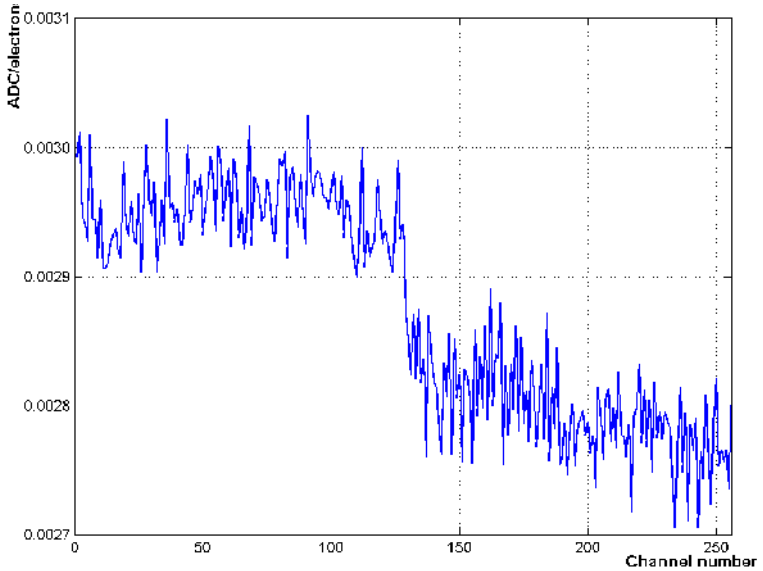

Figure 7.19. *ADC counts per electron (gain) versus channel number for two Beetle chips with no detectors connected. Data acquired at 20 ºC.*

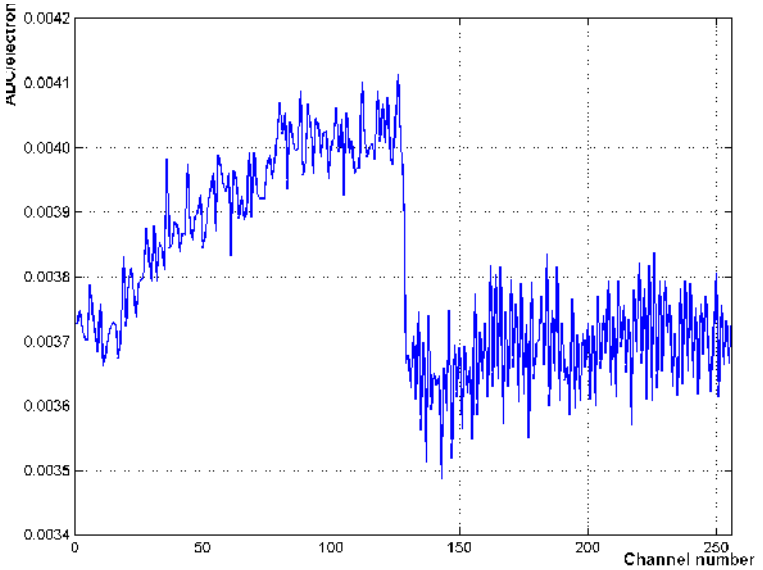

Figure 7.20. *ADC counts per electron (gain) versus channel number for two Beetle chips with no detectors connected. Data acquired at -20 ºC.*

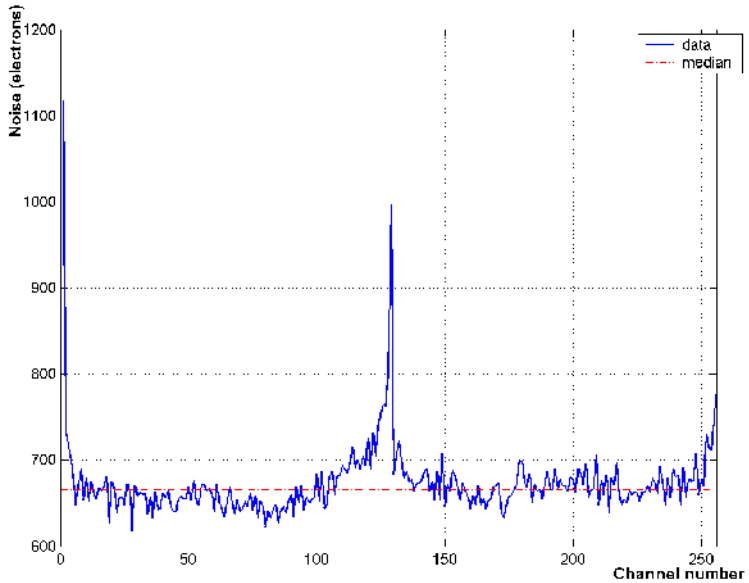

Figure 7.21. *Noise level in electrons versus the channel number for two Beetle chips with no detectors connected. Data acquired at 20 ºC.*

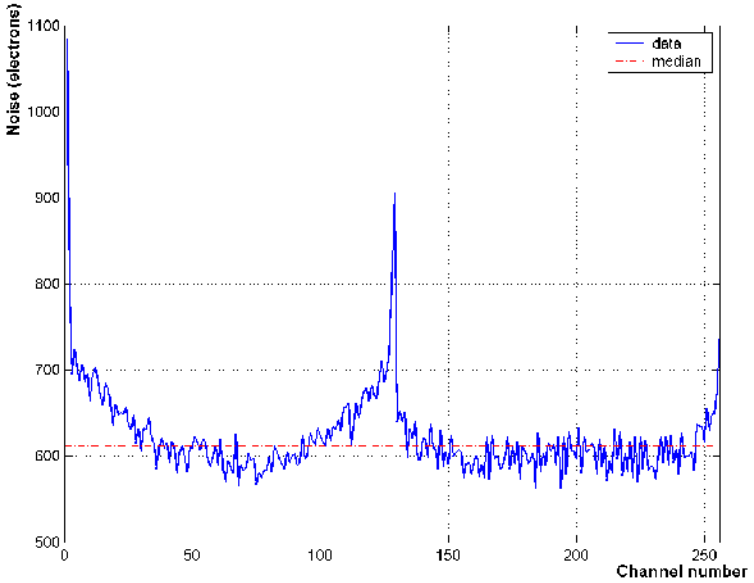

Figure 7.22. *Noise level in electrons versus the channel number for two Beetle chips with no detectors connected. Data acquired at -20 ºC.*

Noise measurements have been also acquired with this new daughter board at

room temperature and at -20ºC, since the calibration data taken at -20ºC can be used. Therefore, the noise level in electrons at -20 ºC can be compared directly with the noise level in electrons at 20 ºC. The noise value at 20 ºC, as the rms value of the pedestal distribution obtained for this daughter board, versus the channel number is plotted in figure 7.21. The median for the data is also represented. The same plot is represented in figure 7.22 for the data obtained at -20ºC. It can be seen that the median of the noise at 20ºC (about 665 electrons) is higher than the median of the noise at -20ºC (about 611 electrons), so the modification on the daughter board analogue buffers does not affect the noise.

7.4.2. Pitch adaptors design

The pitch adaptors design has been modified at the CNM (*Centro Nacional de Microelectrónica*) in order to add an additional track for biasing the detector through the pitch adaptor, as it can be seen in figure 7.23 for the chip pitch adaptor (*Fan_chip*). This bias track has also been implemented in the intermediate pitch-adaptor (*Fan_int*) and the detector pitch-adaptor (*Fan_det*). In the older version of the pitch adaptors, this track was not included so the detectors were biased by connecting two wires from the daughter board HV levels to the detector board.

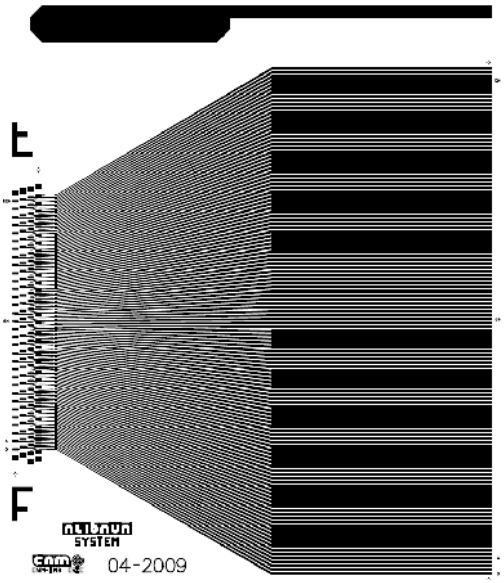

Figure 7.23. *Detailed schematic diagram of the new version of the chip fan-in.*

7.4.3. Chip masking

The system had to work using both Beetle chips continuously, even if only one detector was connected at the daughter board. Therefore, the mother board FPGA logic has been modified in order to address independently each Beetle chip. Thus, the system can work with only one Beetle chip being active for configuration, calibration and acquisition. Time can be saved while configuring and reading data if just one chip should be used.

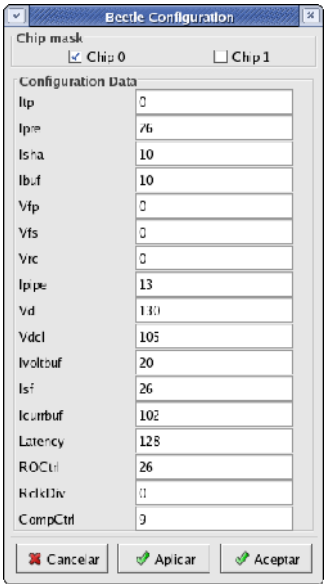

Figure 7.24. *Beetle parameters window of the software. The chip mask can be selected by the user on the top of the window.*

Two FPGA custom logic block have been modified to implement this new feature. In particular, new logic has been added to the *Beetle Slow Control* block to made the Beetle chips addressable independently by means of the bus CHIP(1:0) from the *Microblaze* [101] embedded microprocessor. Moreover, the *Arbiter* block has been modified to accommodate the register SC_CHIP(1:0) in the write-only (*i.e.* the register only can be written by the embedded processor) 32-bit register CONTROL1.

The *Microblaze* processor firmware has been also modified. In particular, the *Beetle Configuration* state has been modified to perform the chip masking configuration. This chip masking is specified by the user from the software by means of a 8-bit chip masking data. The chip masking configuration is maintained

in the *Calibration* and *Acquisition* states for reading the data according to this configuration.

The software has also been redesigned to implement the chip masking. Thus, the Beetle parameters window (figure 7.24) has two selection buttons on the top in order to choose which chips will be active.

7.4.4. Header sampling

The output data of each Beetle chip consists of a 16-bit header and the 128 analogue channels multiplexed. Both the header bits and the analogue channels are 25 ns wide. The system digitalized only the analogue channels and it rejected the header bits. However, the header bits can offer real time information about the Beetle chip status, like the parity of some configuration registers or the pipeline column number of the data read. Therefore, the FPGA logic has been modified in order to digitalize the header as well as the analogue channels.

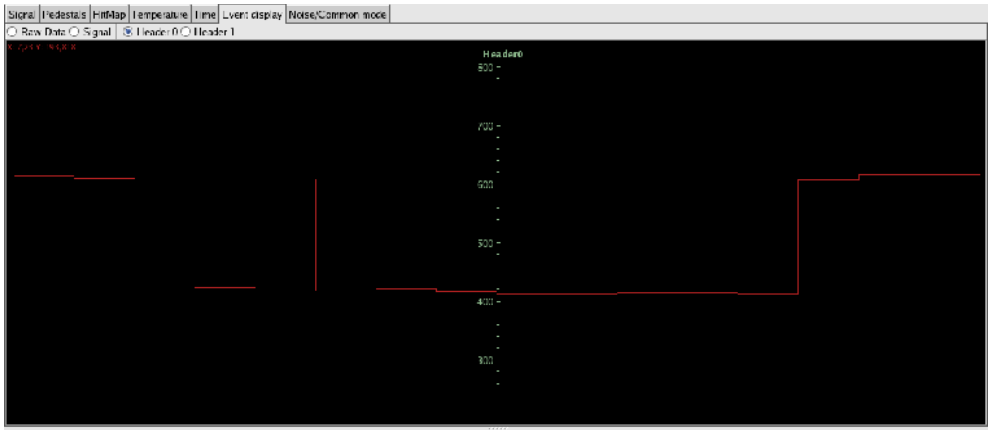

Figure 7.25. *GUI of the software during a pedestals run with the 'Event display' screen selected. The header data from one Beetle chip is shown.*

Some FPGA logic have been modified. In particular, two new FSL links [106], *ADC_input_h0* and *ADC_input_h1*, have been added to the embedded processor to read the header data from two Beetle chips. These two FSL links are asynchronous FIFOs of 16x32 bits configured as slaves (*i.e.* the data are read from the processor). The *ADC Control* blocks have been also modified to generate the sampling clock not only for the 128 analogue multiplexed channels but also for the 16-bit header. Moreover, the new slave FSL links have been connected to the *ADC Control* blocks in the *Arbiter* block.

The *Microblaze* processor firmware has been also redesigned. In particular, the data corresponding to the header are read as well as the data corresponding to the analogue channels in the *Calibration*, *Acquisition* and *Reading* states. The data structures implemented in the firmware to manage the acquired data from the Beetle chips have been modified to handle the header data as well.

The software has been redesigned to monitor the header data and to store these data. For instance, the *Event display* screen (figure 7.25) has two selection buttons to monitor the header data from each Beetle chip. Moreover, the header data is stored together with the data corresponding to the analogue multiplexed channels.

7.4.5. Calibration delay scan

The calibration of the system is performed by injecting pulses on the input channels of the Beetle chips. The amplitude injected in electrons is configured in the Beetle registers. The sign of the injected pulse is changed every time a new pulse is injected. This sign also alternates with the channel number. The pulse is injected when the *Testpulse fast control* signal is activated and the data are read when the *Trigger fast control* signal is activated. The delay between these two signals was fixed. Therefore, Beetle front-end analogue pulse shape could not be reconstructed at calibration stage.

The FPGA logic has been redesigned to be able to reconstruct this Beetle front-end pulse shape from the calibration data. Thus, the delay between *Testpulse* and *Trigger* can be configured by the user with a resolution of 1 ns and a range of 255 ns. In particular, the *Fast Control* block includes a new logic block to delay the *Testpulse* signal with regard to the *Trigger* signal. This is done by combining the carry chain FPGA resources for implementing a fine delay (1 ns) and a 200 MHz clock for coarse delay (5 ns). For the nominal zero delay there is a delay of 133 clock cycles of 25 ns (the nominal latency implemented in the Beetle chip plus five clock cycles) in order to account for the intrinsic delay. The buses DELAY_COARSE(5:0) and DELAY_FINE(2:0) as well as the signal DELAY_CAL have added to the *Fast Control* block to program the delay to implement from the *Microblaze*. Thus, the *Arbiter* block has been modified to accommodate these registers in the write-only (*i.e.* the register only can be written by the embedded processor) 32-bit register CONTROL3.

The *Calibratio*n state of the embedded processor firmware has been changed in order to program the delay to implement as well as the charge to inject for each calibration *Testpulse*. The software has been also redesigned to configure this delay

and this charge, and also to monitor the calibration scan. In the calibration parameters window (figure 7.26), the delay to be implemented can be configured by the user as well as the charge to be injected. Only the charge (with a fixed delay) or the delay (for a fixed charge) can be scanned at a time. The type of scan is selected with the radio buttons at the last row of the window. The value of the fixed variable is set with the first two rows (*Delay* and *Charge*).

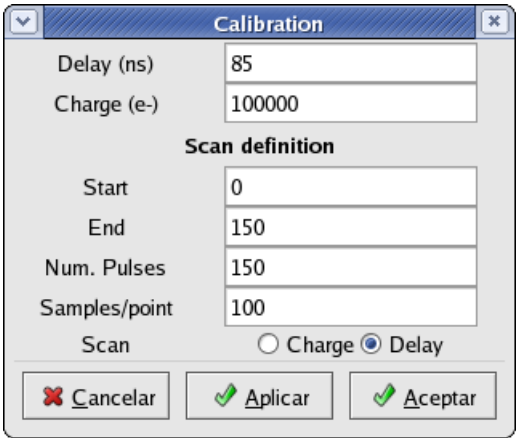

Figure 7.26. *Calibration window with the main parameters that can be set to configure the calibration scan.*

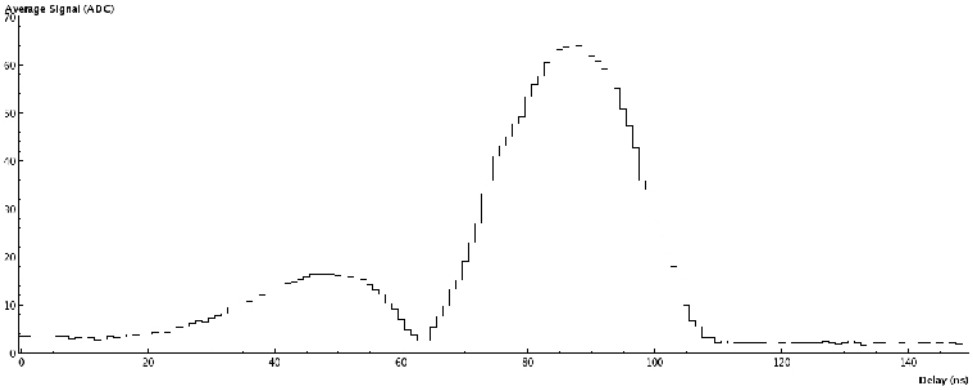

Figure 7.27. *Calibration scan for an injected charge of 24000 electrons and a delay range of 150 ns. The average signal in ADC counts is represented versus the delay in ns.*

The calibration scan can be monitored at the *Signal screen* of the software while performing a calibration run. The content of the *Signal screen* is shown in figure 7.27 for a calibration scan where the charge injected has been fixed to 24000 electrons. The scan has been performed from a delay of 0 ns to a delay of 150 ns in

steps of 1 ns. A hundred of samples have been acquired in each step.

In the figure 7.27 a reconstruction of the Beetle front-end analogue pulse shape can be seen. The pulse starts at about 110 ns and the peak of the pulse (about 66 ADC counts) is at about 85 ns. Therefore, a peaking time of 25 ns can be seen.The signal is represented in absolute values so the subsequent undershoot (signal from 65 ns to 20 ns) is shown as positive but it is a negative signal.

7.4.6. Software reset and hardware version

The system needed an improved reset since the reset by software did not work correctly when the system was stuck. The software reset was a firmware reset. A real software reset has been implemented by modifying the FPGA logic. Some logic has been added to the *USB Control* block in order to spy the data sent by the software and to generate a reset query if a special code is detected. The *Clock Generator* block has been modified to generate a reset signal for all the FPGA logic in case of a software reset is generated, so the reset is done by hardware.

FPGA xc3s400-5pq208 device utilization			
Logic utilization	**Used**	**Available**	**% Utilization**
Slices	3435	3584	95 %
Slice Flip-Flops	1869	7168	26 %
4 input LUTs	3919	7168	54 %
BRAMs	16	16	100 %
Mult 18x18	3	16	18 %

Table 7.3. *Summary of the FPGA resources consumed.*

Finally, the *Reset* state of the *Microblaze* firmware has been redesigned to send a hardware version code to the software in order to load the correct version of the software. In the table 7.3 there are summarized the FPGA resources consumed by the logic implemented in FPGA, including the embedded system.

Chapter 8

Upgrade of the mother board for test beam measurements

The upgrade of the mother board in order to include it as a part of a telescope for test beam measurements is explained in this chapter. The architecture of this telescope is detailed in section 8.1. The description of each part of the telescope is also treated in this section. In section 8.2, the upgrade of the mother board for the telescope is explained, with special emphasis on the FPGA logic and the embedded processor firmware upgrade. The development tests carried out to validate the mother board upgrade are also described in this section.

8.1. Telescope architecture

There is a need of testing silicon detectors in a test beam environment. Therefore, the mother board of the system has been upgraded in order to use it as a part of the readout of a telescope system for test beam measurements. A telescope is a tracking system used to measure, among other properties, the spatial resolution performance when developing new detectors. The intrinsic spatial resolution of silicon microstrip detectors depends on different parameters as the silicon detector characteristics, the readout pitch, the presence of intermediate floating strips, the coupling with the readout electronics and the noise level of the readout electronics. The spatial resolution of a new detector can be determined by means of a detailed study in a test beam. The experimental conditions in a test beam facility are more similar to a high energy physics (HEP) experiment than the conditions in the

laboratory.

The radiation environment in current and future HEP experiments is very aggressive for silicon detectors. For example, fluences up to 10^{15} 1MeV n_{eq}/cm^2 are envisaged for the innermost silicon strip layers of the planned High Luminosity LHC (HL-LHC) detectors [124]. The consequences of this radiation levels in silicon detectors, for instance the strong increase of the full depletion voltage, the high charge trapping and the rising noise, become a key issue on their performance. Thus, it is necessary to develop more robust position detectors for SLHC. Another requirement is to measure important parameters of these detectors, like their spatial resolution, signal-to-noise ratio and charge collection efficiency, to characterize them as radiation hard detectors. The main motivation for the development of this telescope is the measurement of these important parameters in new microstrip, pixel or 3D silicon detectors in a test beam setup.

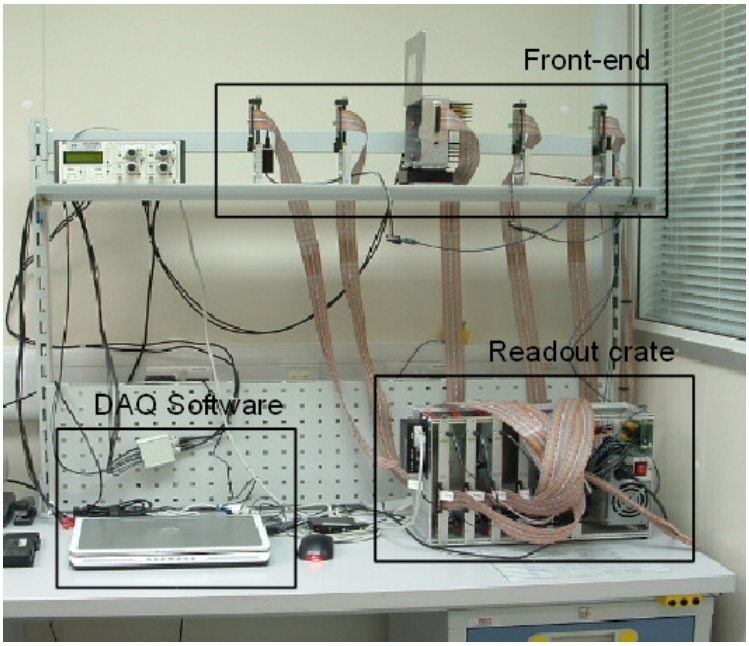

Figure 8.1. *General view of the telescope with its different parts.*

The telescope has several parallel planes with non-irradiated microstrip silicon detectors of fine pitch placed perpendicular to the beam for track reconstruction. The detector under test (DUT), which actually can be different types of irradiated or non-irradiated silicon sensors, is also placed perpendicular to the beam in a centered position with respect to the tracking planes. The silicon detectors are read out by means of Beetle chips. Other detectors can be included in the tracking

planes for obtaining a common trigger signal for the acquisition system. The acquisition system must read the data from the detector under test, the track detectors (x and y coordinates) and the trigger detectors. For this purpose, modified versions of the ALIBAVA mother board are used.

The spatial resolution of the detector under test is evaluated in this telescope from an analogue readout of the charge collected in the detector. The telescope has a front-end part, a readout part and a DAQ software, as it is shown in figure 8.1. The front-end (figure 8.2) has four xy measurement and trigger planes perpendicular to the beam track, which would follow the z coordinate. A XYT board is located in each one of these measurement planes. These four XYT boards are used to reconstruct the beam track and as trigger sources optionally. Four XY points are enough to have precise information of each track with sufficient resolution to correctly analyze the performance of the DUT. There are two scintillators connected to two photomultiplier tubes (PMT) placed in the outermost xy planes for triggering purposes. The detector under test is attached to a DUT board, which is placed in a centered position with regard to the XYT planes. The DUT board is chilled and mounted on a rotary stage. Therefore, the angle between the DUT board and the particle track can be varied. The telescope can accommodate up to sixteen parallel planes, four XYT boards and up to twelve DUT boards.

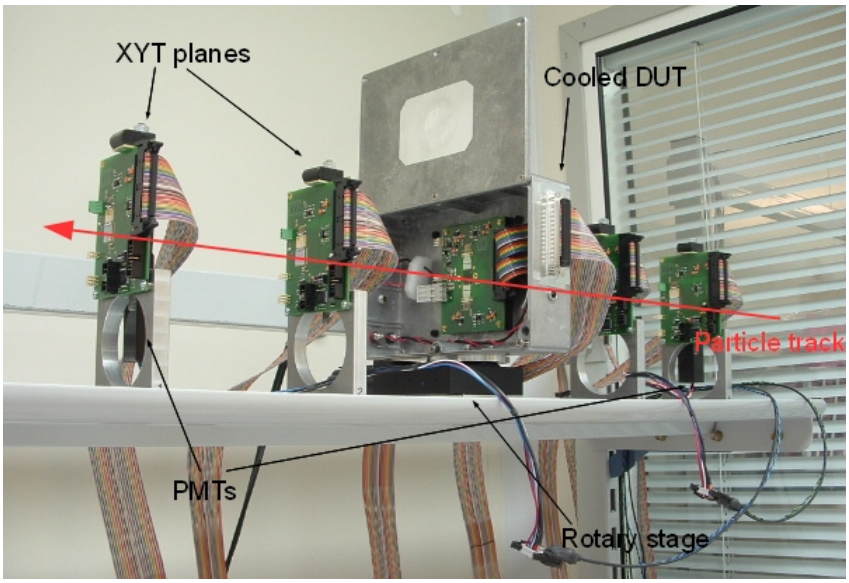

Figure 8.2. *Front-end part of the telescopew. The base unit, four XYT boards, a cooled DUT, two scintillators, two PMT and the rotary stage are shown.*

Each XYT or DUT board is read out by an ALIBAVA mother board (so-called slave board) upgraded to operate correctly in the telescope. All the slave boards are connected to a master board using a parallel bus with a custom digital protocol. The master board collects the data and distributes the common control signals (*Trigger*, *Clock* and *Event reset*) for the slave mother boards. This board is connected with the DAQ (*data acquisition*) software via Ethernet [125]. The readout part of the system is shown in the figure 8.3. This part includes five slave mother boards, the master board, the power supply system and the driver for the DUT rotary stage. It is located in a crate with a standard PC-like power supply. The power supply provides 5 V and 12 V for powering all the telescope hardware except for the silicon detectors of the XYT and DUT boards. The detectors of these boards are biased with independent power supplies.

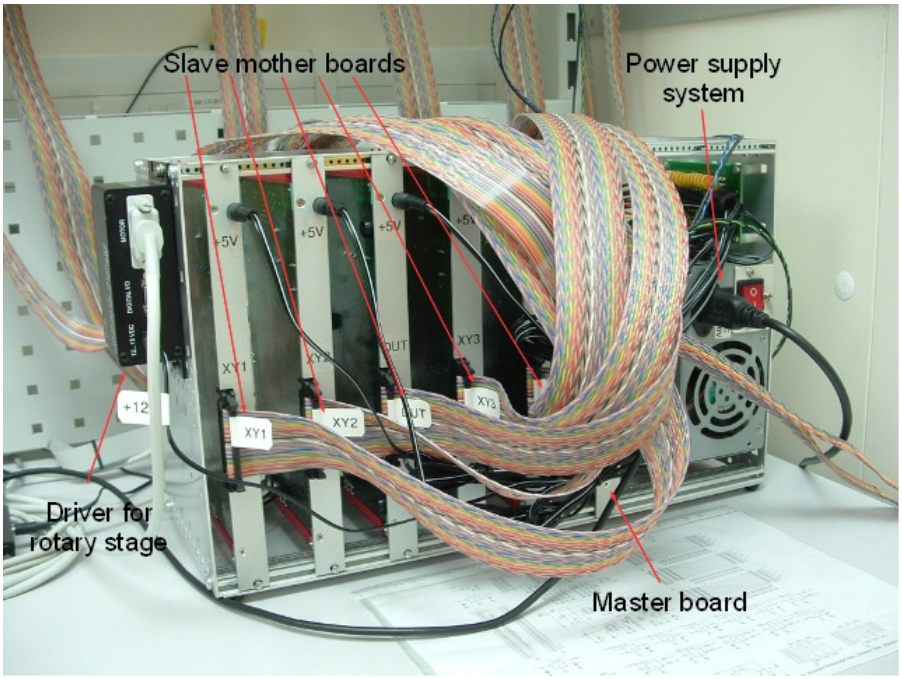

Figure 8.3. *Readout part of the telescope. The crate, five slave boards, the master board, the power supply system and a driver for the DUT rotary stage are shown.*

The DAQ software collects and processes the data acquired to store them in an adequate format for further off-line analysis. It also monitors the telescope to control its operation.

8.1.2. DUT board

The DUT boards use two Beetle chips for the readout of a DUT, either strip or fan-out pixel silicon detectors of different types. The block diagram of the DUT board is shown in figure 8.4 whereas a picture of the DUT board is shown in figure 8.5. The Beetle chips operate configured as in the daughter board (see chapter 4). A parallel configuration for the Beetle chips has been considered. There are two different analogue data lines, one for each Beetle chip, but the fast and slow control lines are shared by both chips. Each Beetle chip has its own I^2C address. The differential current output signal of each Beetle chip is closed with 100 Ω and the voltage is buffered with a line driver with the same design as the one in the daughter board. This buffer can provide line equalization by adding two capacitors if the cable is longer than 10 m [51].

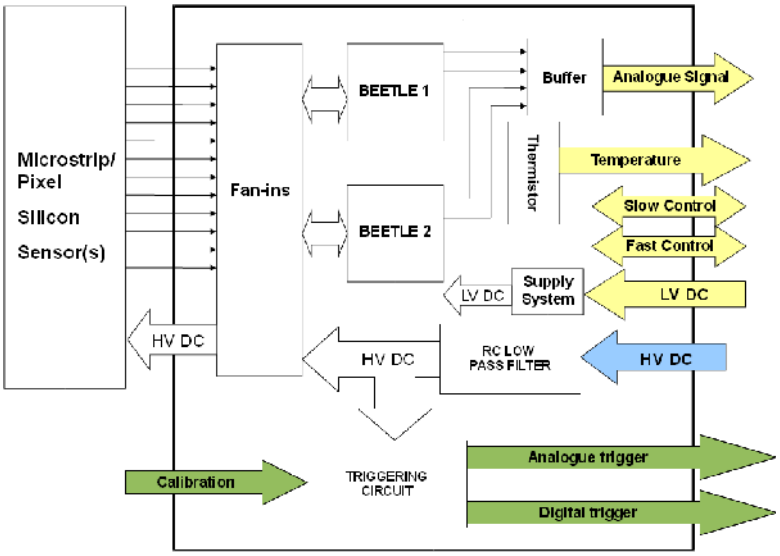

Figure 8.4. *Block diagram of the DUT board.*

The board can operate in a self-triggering mode providing analogue and digital pulses for the readout system, since it includes a trigger circuit from the detector back-plane signal. The circuit consists of a charge sensitive amplification (CSA) stage, a constant fraction discriminator (CFD) and a charge injection stage for calibration. The output signal of the CSA is a pulse with a fixed width of 60 ns, which can be used externally for triggering purposes or discriminated on-board using the CFD to obtain digital pulses. The board features a temperature sensor attached to the DUT back-plane via the copper chuck.

The DUT board is power supplied from a 5V *DC* level sent from the corresponding slave board. This supply level is regulated by five LDO (Low Drop Out) linear regulators to obtain the *DC* supply levels required by the chips (2.5 V), the line drivers (4.5 V) and the trigger circuit (4.5 V). The detector high voltage bias is supplied directly from a independent power supply to control the current provided. This supply level is decoupled prior to the detector.

The detector is placed in the middle of the DUT board. The back-plane of the detector is glued to the metallic chuck, which is attached to the board with screws. Two pitch adaptors are used to connect the detector strips or pixels to the Beetle chips. The pitch adaptors have pads repeated 10 times at the detector side to reuse the DUT board with different detectors. The DUT board can be mounted on a rotary stage controlled manually or by software. A peltier element is used for cooling the DUT board. The heat sink of the Peltier element is cooled by means of two air fans. The Peltier element is controlled by a USB temperature controller.

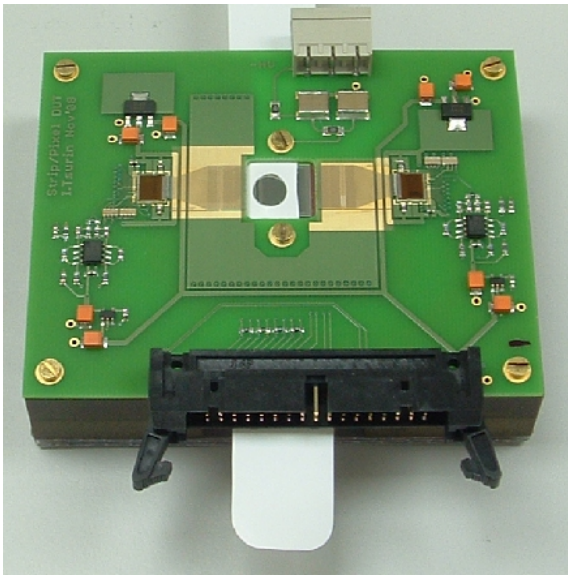

Figure 8.5. *Picture of a DUT board prototype without the self-triggering circuit.*

8.1.3. XYT board

The XYT board has been designed to measure the track space points by using two silicon strip detectors mounted back-to-back at 90 degrees. The block diagram of the XYT board is shown in figure 8.6 and a picture of a board prototype is shown in figure 8.7. The coordinate measurements are carried out with two Beetle

chips. These chips have the same configuration as the ones in the DUT board. They work in parallel mode and they have two output analogue multiplexed signals. The *fast and slow control* signals are shared by the chips. The buffer stage is also equal to the DUT board buffer stage.

The board can also trigger on the particle tracks in the test beam using a trigger circuit connected the detector back-plane signal. This circuit is the same as the one of the DUT board. The board includes a CPLD (*complex programmable logic device*) with the logic necessary to allow for the synchronization of the trigger signal generated on-board to a common clock frequency, delaying and implementing coincidence with other XYT cards. The digital trigger signal from the trigger circuit is connected to the CPLD on each XYT board. The trigger signal processed in the CPLD is connected to an external bus through a 14 way header connector. The input trigger signals from other XYT boards, the common clock signal from the master board and the power supply level for the CPLD are also obtained from this external bus. This bus is connected to the XYT boards and to the master board. The CPLD can be programmed on-board by JTAG (*joint test action group,* standard IEEE 1149.1) through a dedicated connector.

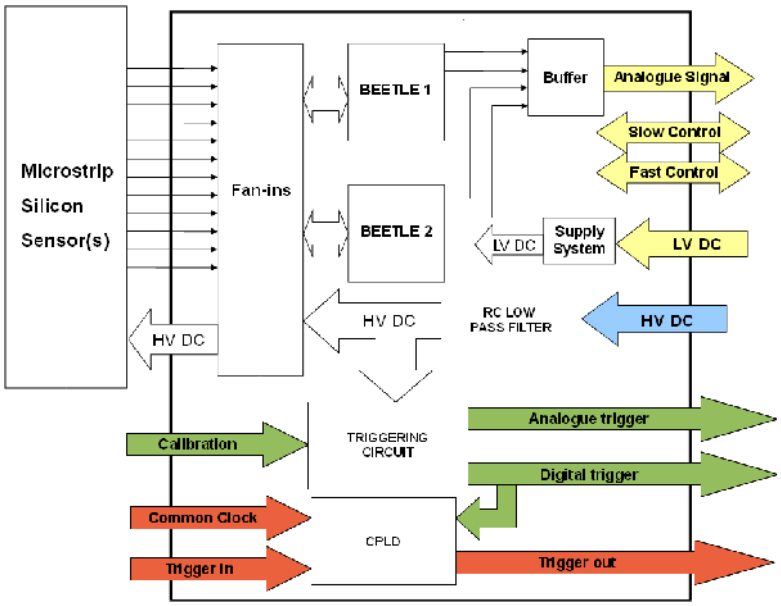

Figure 8.6. *Block diagram of the XYT board.*

The XYT board is powered from a 5 V *DC* level sent from the corresponding slave board. LDO linear regulators are used to obtain the *DC* supply levels required by the chips (2.5 V), the line drivers (4.5 V) and the trigger circuit (4.5 V). The

CPLD is powered through a LDO from a 5 V *DC* level sent from the master board. The detector high voltage bias is supplied directly from a independent power supply to control the current provided. This supply level is decoupled prior to the detector.

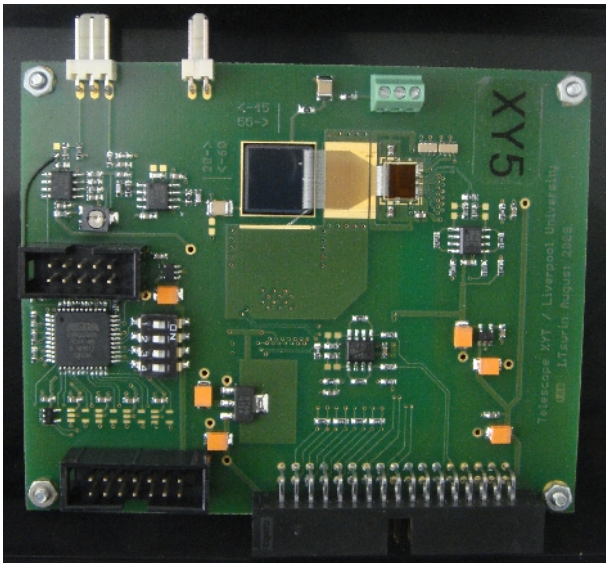

Figure 8.7. *Picture of a DUT board prototype*

8.1.4. The slave board and the readout bus board

A slave board, a modified version of the ALIBAVA mother board, is used to read out and to control each XYT/DUT board from a common trigger signal and clock signal. The mother board is widely described in chapter five. The mother board hardware has been slightly modified whereas a deeper redesign has been carried out with the FPGA logic and the embedded processor firmware. The redesign of the FPGA logic and the firmware is treated with detail in section 8.2. The block diagram of the slave board is depicted in figure 8.8 while a picture of the slave board is shown in figure 8.9.

The former trigger inputs of the mother board are used in the slave board for the common input signals (*Clock*, *Trigger* and *Event reset*). The common *Clock* signal has a frequency of 40 MHz. These digital low level (110 mV at high level) signals are discriminated by three fast leading-edge discriminators with an input impedance of 51 Ω, since originally they were intended either for fast analogue or current inputs. The discrimination thresholds are generated by a 12-bit DAC

controlled by the FPGA. The TDC on board is used to have a time stamp of each input trigger with respect to the common *Clock*, so there is no need of synchronization between the *Trigger* and *Clock* signals. This TDC has a resolution of 600 ps and a range of 25 ns for this application.

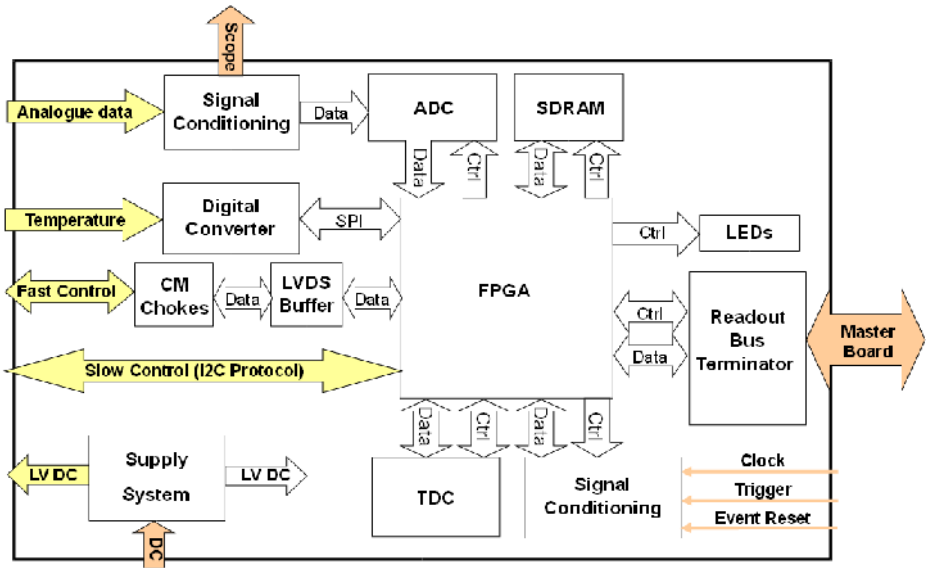

Figure 8.8. *Block diagram of the slave board*

The data collected by each slave board are sent to the master board by means of a local data/address bus following a custom digital protocol. The former USB controller has been replaced in each slave board by a digital connector and a readout bus terminator for this digital communication. The bus has 8 lines for bidirectional data, 4 lines for data control and 16 lines for the slave board addressing (up to 16 slave mother boards can be addressed). The master board and the slave mother boards are connected to the same bus through their readout bus terminators. The readout bus terminator is a small board which replaces the USB controller bus keeping the protocol of the USB controller for the communication with the FPGA.

The rest of the hardware blocks operate as in the original mother board with the exception of the Trigger Output blocks, which are not used in the slave board. The slave board board operates from a 5 V *DC* input level. From this supply rail digital and analogue power supplies for the on board hardware are generated by means *DC-DC* converters and a LDO regulators. The DUT/XYT board power supply (5 V) is generated by a *DC-DC* converter.

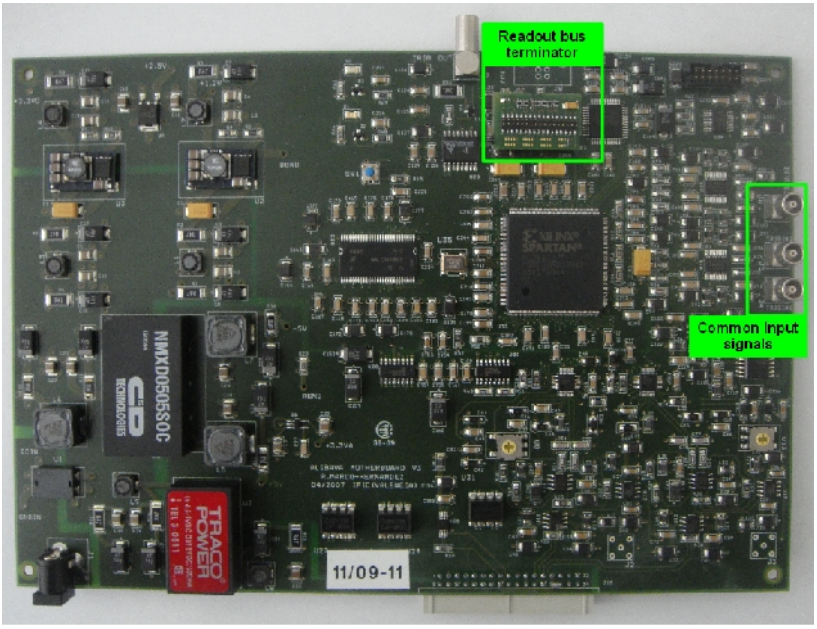

Figure 8.9. *Picture of a slave board with a readout bus terminator connected.*

8.1.5. The master board

The master board is implemented with a Xilinx Spartan 3E development board [126] and a custom patch board. A picture of the master board is shown in figure 8.10. The board distributes the common *Trigger*, *Clock* and *Event reset* signals to the slave mother boards through the patch board where the corresponding cables are connected. The *Trigger* signal is generated by the system either from the XYT boards trigger signals or from two external PMTs. In both cases the patch board provides the connections and circuitry required by these signals. A CFD stage is included for each PMT signal, whose digital output is connected to the FPGA. The bus of the trigger signals from the XYT boards is terminated in the patch board, with LVDS receivers for the trigger input signals and a LVDS driver for a synchronization clock. The incoming trigger signals are connected to the FPGA. The common clock and event reset signals are produced directly by the FPGA.

The master board also manages the data streams coming from up to sixteen slave mother boards. The patch board provides a connector and terminations for the digital data bus from the slave mother boards. A decoder circuit has been also included for addressing the Alibava boards with a 4-bit bus. The 8-bit data bus, the 4-bit data control bus and the 4-bit address bus are connected to the FPGA.

Moreover, the master board performs the communication with the DAQ software via 100 Mbits/s Ethernet. An Ethernet controller has been implemented in the board FPGA for this purpose. Therefore, the board behaves as a communication bridge, addressing and collecting the data from the corresponding slave board and sending these data in the corresponding Ethernet frames to the software.

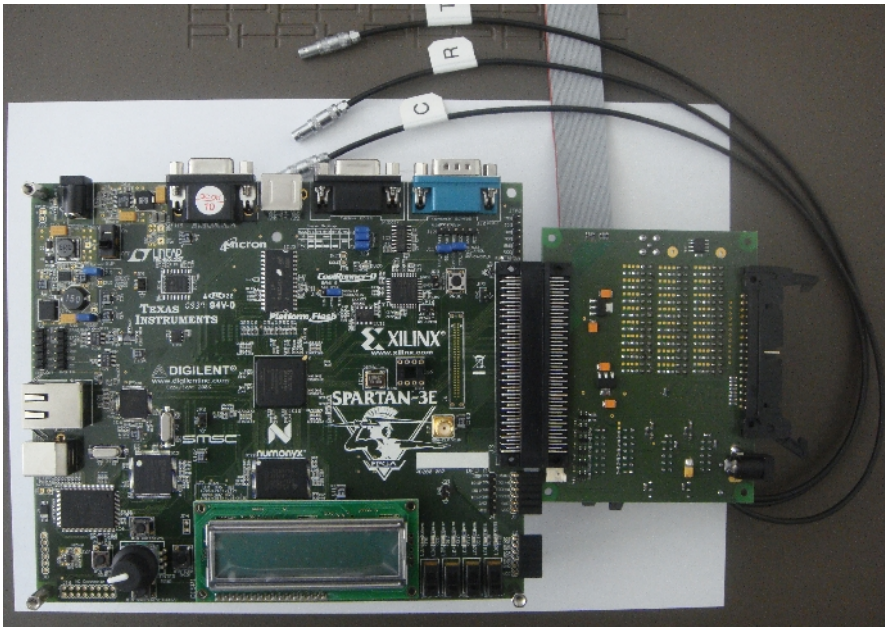

Figure 8.10. *Picture of the master board implemented with a Xilinx Spartan 3E development board connected to a custom patch board.*

The board has also a test channel for testing in a standalone mode either a XYT or DUT board. The required *fast* and *slow control* signals for the Beetle chips are generated by the FPGA. For the analogue multiplexed signals from the Beetle chips, a buffer stage has been included to connect these signals to a scope. New functions can be added in the future to the master board by adding new digital logic to the FPGA board.

8.1.6. Power supply distribution

The power supply distribution for the system is carried out by a small power supply which supplies 5 V and 12 V from the mains. There is a power distribution board connected to the power supply to provide these levels to the slave mother boards, master board, PMTs, DUT rotation stage and air coolers for the Peltier

element of the DUT.

8.1.7. The DAQ software

The DAQ software controls the hardware of the telescope by means of codes which are sent by raw Ethernet communication. These codes are distributed to the corresponding slave board by the master board. The slave mother boards also send to the DAQ software, through the master board, the data acquired and status codes informing about the state of the hardware. The master board acts as a communication bridge by addressing the corresponding slave board and implementing the Ethernet communication with the software. Therefore, the DAQ software can control the hardware for configuration, calibration and data acquisition. Moreover, the software processes the data acquired, which are sent in a raw format from the slave mother boards, in order to obtain data with physical meaning. The DAQ software has been designed using two levels. The low level implements the communication between the software and the master board by raw Ethernet and the low level data processing. The high level includes the Graphic User Interface (GUI) and output file generation. The software is being developed using C++ language and it is based on the software designed for the standalone system (daughter board and mother board).

8.1.8. Silicon detectors and track fitting software

The detectors of the XYT boards can be *n*-type or *p*-type. A thickness of 300 μm can be used for obtaining robust signal amplitudes and performing charge sharing. Either 80 μm pitch detectors or 40 μm pitch detectors with intermediate strips, for better spatial resolution, can be used. The detectors are connected to the Beetle chips by means of pitch adaptors. The DUT board can accommodate different types of non-irradiated or irradiated silicon detectors, like microstrip, pixel or 3D silicon detectors. Positive and negative current input signals can be read. The board can read up to 256 input channels and pitch adaptors are used to connect the detector(s) to the Beetle chips.

Track fitting software is also being developed for the data obtained with the DAQ software. The charged particles leave a track of hits or detector measurements along their path in the telescope. The track fitting is based in the minimization of the track-hit residuals:

$$\boldsymbol{r} = \boldsymbol{m} - \boldsymbol{e} \quad (8.1)$$

where $\boldsymbol{r}$ represents the residual vector of the measurements of a track, $\boldsymbol{m}$ is the vector of measurement of this track in the different sensors and $\boldsymbol{e}$ their expected values according to the test track parameters $\boldsymbol{\tau}$.

The track fitting consists in finding the set of track parameters ($\boldsymbol{\tau}_0$) which minimizes the following χ^2:

$$\chi^2 = \boldsymbol{r}^{\boldsymbol{T}} V^{-1} \boldsymbol{r} \quad (8.2)$$

being V the covariance matrix of the detector measurements.

Of course, the accuracy which the track parameters can be determined depends on the precision of the measurements (through the covariance matrix V). But also important is to make sure that no bias affects the track parameters. That depends on how well the measurements ($\boldsymbol{m}$) represent the real path of the particle and no errors or deformations of the detector affect the measurements positioning.

8.2. Mother board upgrade

The ALIBAVA mother board has been modified to integrate it as a slave board in the telescope. The slave board is in charge of reading the data acquired with the corresponding DUT or XYT board as well as controlling the corresponding DUT/XYT board. This is carried out using three common signals (*Trigger*, *Clock* and *Event reset*) sent from the master board to the slave mother boards in order to have a synchronized operation of the system. The hardware modifications carried out to the mother board have been described in section 8.1.5. These are minor modifications and the most of the board hardware is used without any change. However, the most part of the upgrade of the mother board have been performed in the FPGA logic and the firmware of the embedded processor.

8.2.1. FPGA logic

The FPGA logic of the slave board is based on the FPGA logic of the original mother board with the improvements described in chapter 7. The block diagram of the slave board FPGA logic is depicted in figure 8.11. It is a combination of custom VHDL blocks for controlling the hardware and an embedded system. The embedded system includes a 32-bit *Microblaze* software core processor and a SDRAM controller as a peripheral. The interface between the embedded system and the custom logic blocks consists of eight FIFOs integrated in the *Microblaze* and a custom *Arbiter* block. The *Arbiter* block implements the required logic to

manage the FIFOs and the signal interfaces of the custom logic blocks.

The *external signal generator* block (ESG) processes the common *Trigger* and *Event reset* signals from the master board in order to generate valid EXT_TRIG and EXT_RST signals for other FPGA blocks. It also produces internal clocks of 40 MHz (EXT_CLK_40) and 200 MHz (EXT_CLK_200), by means of a on-chip *Digital Clock Manager* (DCM), from the 40 MHz common *Clock* signal coming from the master board.

The *ADC Control* block reads the digitized data frames when the corresponding *DataValid* signal is active and it stores these frames in an internal FIFO memory. These frames correspond to the header and analogue multiplexed channels coming form the Beetle chips and digitized in the on board ADC. The *Master Communication* block (MC) performs the communication with the master board by means of a 8-bit parallel data bidirectional transmission and a 4-bit data control. This protocol is the same as the one used in the *USB control* block of the mother board.

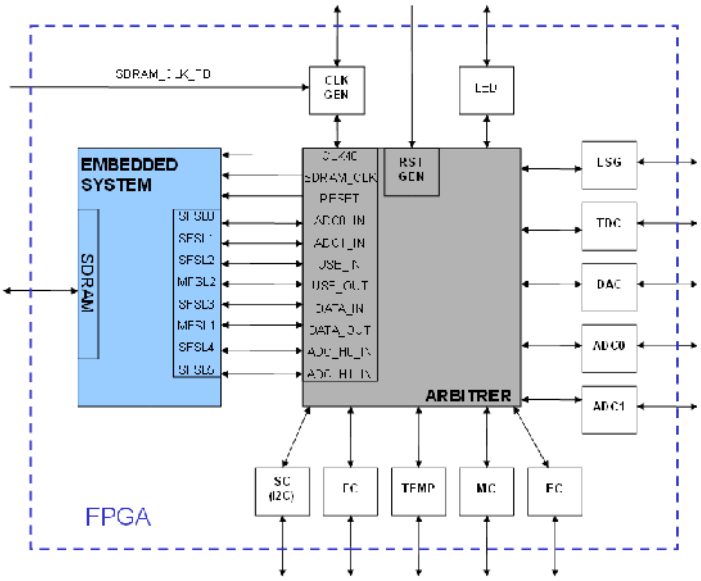

Figure 8.11. *Block diagram of the logic implemented in the FPGA of the slave board. This diagram can be compared with the orginal one of the ALIBAVA mother board (Figure 5.49).*

The *Slow Control* block (SC) includes an I2C bus controller to program the Beetle configuration registers. The *Fast Control* block (FC) generates the LVDS output signals (*Clk*, *Reset*, *Trigger* and *TestPulse*) for the Beetle *fast control*. The

Clk is produced from EXT_CLK_40 clock. The *Trigger* signal is generated after a common *Trigger* has been received from the master board, *i.e.* an EXT_TRIG pulse is produced, taking into account both the Beetle analogue pipeline latency (nominally 128 *Clk* cycles) and the particular synchronization delay. The *TestPulse* is generated from an internal calibration signal. Both blocks are the same as the ones implemented in the mother board.

The *TDC Control* block controls the on board TDC, which measures the time from the leading edge of a start signal to the leading edge of a stop signal. The start signal is generated by the FPGA from the EXT_TRIG pulse signal which will be active when a common input *Trigger* is detected. The stop signal is also generated by the FPGA from the EXT_CLK_40 clock, if a start signal has been generated previously.

The *Event Counter* block (EC) implements a counter which value is increased for each data event acquired, *i.e.* with each EXT_TRIG pulse. The counter is reset either with the internal reset signal or the event reset signal (EXT_RST).

The *Clock Generator* block (CLK_GEN) produces an internal reset signal either from an external hardware reset or a software reset. It also generates internal clocks of 40 MHz (CLK_40) and 200 MHz (CLK_200) from the 40 MHz on board oscillator clock signal. Only the *Fast Control* block operates with the external clocks, *i.e.* all the Beetle chips are synchronous to the common *Clock* signal coming from the master board. The rest of the blocks work with the on-board clocks (CLK40 and CLK200).

The rest of the blocks (*Temperature Control*, *LED Control* and *DAC Control*) are the identical as the ones implemented in the mother board FPGA. In the table 8.1 there are summarized the FPGA resources consumed by the logic implemented in FPGA, including the embedded system. It can be seen that there is room to accommodate more logic if required.

FPGA xc3s400-5pq208 device utilization			
Logic utilization	**Used**	**Available**	**% Utilization**
Slices	3365	3584	93 %
Slice Flip-Flops	1797	7168	25 %
4 input LUTs	3851	7168	53 %
BRAMs	16	16	100 %
Mult 18x18	3	16	18 %

Table 8.1. *Summary of the logic resources consumed in the Xilinx xc3s400-5pq208 FPGA of the slave board.*

8.2.2 FPGA firmware

The embedded processor firmware of the slave board has been designed as a finite state machine (FSM). It is based on the former processor firmware of the mother board with the improvements described in chapter 7. The functionality of the system have been defined in this FSM, whose states are depicted in figure 8.12. The embedded processor firmware is included in the FPGA bitstream since its memory size is not very large and it is stored in the on board PROM with the FPGA logic bitstream.

The FSM main state is a *Waiting* state. The FSM waits for an order coming from the PC software by means of the MC block. Depending on the order, the FSM will choose among the rest of states as it is shown in figure 8.12. When the system is powered on or after an external reset, the FSM goes to the *Reset* state prior to the *Waiting* state. In this state, all the system is initialized. After the FPGA configuration, the system also goes to this state. In the *Beetle Configuration* state, the configuration registers of the Beetle chips are programmed by *slow control*. In the *Trigger In Configuration* state, the DAC voltage thresholds are programmed to discriminate adequately the input common signals from the master board.

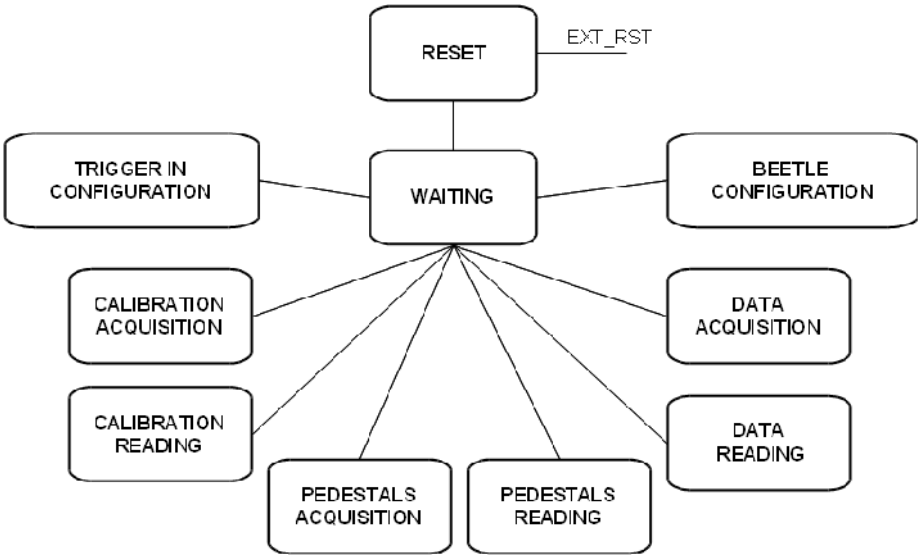

Figure 8.12. *States diagram of the finite state machine (FSM) implemented in the embedded processor firmware.*

The acquisition states have the same structure but different data types are acquired and stored in the on board SDRAM. In the *Calibration Acquisition* state,

the system is calibrated by the Beetle internal test pulse generator, *i.e.* known amplitude readouts are acquired in order to have calibration data. The *Testpulse* signal sent to the Beetle chips can be delayed with regard to the *Trigger* signal in order to perform a reconstruction of the analogue pulse shape of the Beetle front-end. This delay can be programmed by the user. The timing data (delay implemented), temperature data and the digitized data from the Beetle chips are acquired for each event. In the *Pedestals Acquisition* state, the readout is just used to determine the baseline for each input channel of the Beetle chips. A programmable number of readouts can be acquired. For each event, a readout of the Beetle chips and a temperature readout are stored in the SDRAM. In the *Data Acquisition* state, a programmable number of readouts can be acquired from the external trigger input using the TDC. For each event, a readout of Beetle chips, a TDC readout, a temperature readout and an event number are stored in the SDRAM. In the *Reading* states, the last type of acquisition is read from SDRAM and data are sent to the master board.

8.2.3. Development tests

The aim of the development tests carried out was to verify the correct operation of the FPGA logic and the firmware upgrade of the mother board. Several MATLAB [112] algorithms were developed following the DAQ software specifications to mimic this software behaviour. A mother board with the USB controller was used to debug the new FPGA logic and the firmware since the communication with MATLAB is through a serial port. However, this fact does not affect the to FPGA logic and firmware since the MC block uses the same protocol as the former USB block. A daughter board connected to the mother board was used for the development tests. This daughter board has the same functionality as the DUT board or the XYT board for testing purposes, *i.e.* the Beetle chips have the same configuration and operation mode.

Two pulse generators were used in order to produce the common *Clock*, *Trigger* and *Event reset* signals. The *Clock* frequency was fixed to 40 MHz. The *Trigger* signal frequency was from tens of Hz up to 100 kHz maintaining a pulse width of 100 ns. The *Clock* and *Trigger* leading edges were not synchronized. The signal high level of these signals was fixed to 110 mV and the low level to 0 mV.

The different firmware states were tested successfully. Thus, the Beetle chips and the trigger inputs of the mother board could be configured correctly. Regarding the calibration, calibration data could be acquired using the *Calibration Acquisition* and *Reading* states. Both calibration data with a fixed charge and variable delay and *vice versa* were acquired. The calibration data acquired corresponded to the

charge injected and the Beetle front-end analogue pulse shape could be reconstructed. The *Pedestals Acquisition* and the *Reading* states were also tested by acquiring thousands of events. The pedestals and noise levels obtained were as expected. Finally, the *Data Acquisition* and *Reading* states were tested by taking thousands of events. Since, no detector was connected to the daughter board, the signal obtained in each channel was similar to the pedestals. The TDC measurement obtained for each event corresponded to the time separation between the *Trigger* leading edge and the *Clock* leading edge. The event number was incremented with each event acquired as expected. The temperature readouts in all *Acquisition* states corresponded to the real temperature.

Conclusions

The design, development and implementation of a readout system for microstrip silicon sensors has been reported as well as its upgrade in order to use part of this system, the mother board, as a readout of a telescope for test beam measurements. This system has been developed in the framework of the ALIBAVA collaboration (University of Liverpool, CNM of Barcelona and IFIC of Valencia) which is integrated in the RD50 collaboration. The main motivation for designing this system arises from the need of evaluating different characteristics of the microstrip silicon sensors, like the collected charge or the spatial resolution, in an environment similar to the one envisaged for the HL-LHC, in particular the high irradiation dose and the readout electronics. Therefore, first of all, the framework in which this system has been developed has been stated as an introduction to the work carried out.

Secondly, an overview of the microstrip silicon sensors has been presented. In particular, both the silicon properties and the *pn* junction behaviour have been summarized, including the main *pn* junction characteristics of interest for silicon detectors. Then, the principles of operation of the microstrip silicon sensors have been described. Finally, other aspects as the radiation damage, the associated readout electronics, the noise sources or the detector structures have been addressed.

Thirdly, the motivations for developing this system have been detailed. The specifications for the system have been established according to the constraints imposed by its application. From these motivations and these specifications, the system architecture has been presented and discussed. Thus, according to the system architecture, the system has been divided into two main parts: a hardware part and a software part. Also the hardware part has been based on a dual board system: a daughter board and a mother board.

Fourthly, the structure and components of the different parts of the system have been described in detail. Moreover, the design of the different parts of the system has been discussed in terms of the specifications fulfillment. The daughter board structure and design has been detailed. This board is a small board which contains two Beetle readout chips and pitch adaptors and detector support to interface the

sensors. It also incorporates the hardware required for buffering the analogue data sent to the mother board and for receiving the control and configuration signals for the Beetle chips.

The mother board design has been treated in depth. The mother board is intended to process the analogue data that come from the readout chips as well as to process the trigger input signal in case of radioactive source setup or to generate a trigger signal if a laser setup is used. It is also used to control the whole system and to communicate with a PC via USB. The mother board hardware block diagram has been presented and then the design of the different hardware blocks. Afterwards, the block diagram of the FPGA logic has been also presented, describing the functionality and the structure of all the custom logic blocks. Furthermore, the structure and components of the embedded system included in the FPGA have been described. Finally, the functionality of the firmware programmed in the FPGA embedded processor has been explained in detail. This firmware has been implemented as a finite state machine which incorporates all the possible states of the system.

The main characteristics and the structure of the software have been reported. The function of the software is to control the whole system and processing the data acquired from the sensors in order to store it in an adequate format file. It controls the whole system for configuration, calibration and acquisition. It incorporates a graphical user interface for the communication between the system and the user. It also generates information about the acquisition and stores this information in adequate output file format. Finally, the software should monitor the state of the system in different situations. This software has been divided into two levels according to their functions: a low level software and a high level software. The low level software deals with the data exchange by USB with the mother board and with the low level data processing. The high level software implements the GUI for the communication between the user and the system, the data monitoring as well as the data output file generation.

Fifthly, the development process carried out with the hardware prototype and the software to validate the design has been explained. Moreover, some results corresponding to measurements acquired with the system have been presented. These measurements were done both with a laser setup and a radioactive source setup. Non-irradiated and irradiated n-type and p-type microstrip silicon sensors were used to analyze the system performance showing that the system works correctly.

As a result of the work carried out, a number of systems, 126 daughter boards

and 71 mother boards, have been produced and tested. The production process of the system and the quality tests carried out have been explained. Some systems (34 full systems and 35 daughter boards extra) have been distributed mainly among different research groups of the RD50 collaboration which are interested in using the system for their research lines. Nevertheless, other research groups not integrated in the RD50 collaboration but involved in silicon detectors research have also acquired the system. Some improvements, based on the experience of the users which have taken measurements with the system, have been implemented on the system.

Finally, the upgrade of the mother board in order to use it as a part of the readout of a telescope for test beam has been explained. The motivations for the design of such a system have been presented. Then, the architecture of the telescope has been explained, describing the design of the different parts of the telescope. The upgrade of the FPGA logic and the embedded processor firmware of the mother board for the telescope has been presented. Some development tests performed to validate this mother board upgrade have been also described.

At the present time, the telescope hardware has been produced and tested separately, whereas the software is being developed. The next step would be to finish the development of the DAQ software and the track alignment software of the telescope. Afterwards, it must be accomplished the integration and commissioning of the hardware and the software as well as the data acquisition in a test beam.

Resumen

1. Introducción

1.1. Marco de trabajo

Los grandes experimentos de Física de Partículas tienen una meta común, descubrir de qué esta compuesta la materia y cómo permanece unida. Dado que todo está hecho de materia, esto lleva a la comprensión de las fuerzas fundamentales que gobiernan la materia. Actualmente, se sabe que la materia está compuesta de doce bloques básicos llamados partículas fundamentales (ver tabla 1.1). Estos bloques básicos están gobernados por cuatro fuerzas fundamentales (ver tabla 1.2). La teoría que describe estos bloques básicos y tres de las fuerzas fundamentales se denomina el Modelo Estándar [1] de la física de partículas. Este modelo se ha desarrollado durante el siglo XX y no ha variado prácticamente desde 1974. Desde entonces la Física de Partículas intenta refutar las predicciones del Modelo Estándar así como determinar sus parámetros de forma precisa. También se intenta encontrar las limitaciones del modelo y explorar nueva física más allá del Modelo Estándar.

El CERN (*Conseil Européen pour la Reserche Nucléaire*) [2] es en la actualidad el mayor centro de investigación de Física de Partículas del mundo. El CERN posee un complejo de aceleradores interconectados (figura 1.1) en los que se aceleran haces de partículas progresivamente hasta una fracción por debajo de la velocidad de la luz. El último acelerador es el LHC [3, 4] (*Large Hadron Collider*), el colisionador de hadrones más potente del mundo, que tiene cuatro enormes detectores en sus cuatro puntos de colisión.

El LHC es un colisionador circular de 27 km de longitud por el que se aceleran dos haces de protones (o iones de plomo) en sentidos opuestos, que alcanzan una energía de 7 TeV por haz y una luminosidad de 10^{34} cm^{-1} s^{-1}. Esto permitirá explorar física en la escala del TeV. Los haces no son continuos, sino que están formados por paquetes de protones (o iones de Pb) separados unos de otros. Cada paquete contiene hasta 1.15 x 10^{11} protones. Las colisiones se producen a una

frecuencia de 40 MHz (es decir, hay una colisión cada 25 ns). En la tabla 1.4 se muestran los parámetros más relevantes del LHC. Los objetivos del LHC son realizar medidas de alta precisión de observables ya conocidos del Modelo Estándar, que proporcione respuestas al origen de la masa de las partículas a través del bosón de Higgs, así como buscar pruebas experimentales de nuevas teorías más allá del Modelo Estándar como SuperSimetría o Dimensiones Extra.

El LHC alberga cuatro grandes detectores localizados en los cuatro puntos de colisión del acelerador (figura 1.2), aunque en realidad se llevan a cabo seis experimentos.

- ATLAS (A large Toroidal LHC ApparatuS) [5]: Detector de propósito general que realiza medidas de alta precisión dentro del Modelo Estándar y busca el bosón de Higgs y evidencias de nueva física. ATLAS es un detector de geometría cilíndrica formado por distintas capas concéntricas de subdetectores. El detector de trazas (o Detector Interno) es el sistema más cercano al haz es el detector de trazas y se compone de subdetectores basados en silicio (píxeles y microbandas) y en tubos de deriva. Después se encuentran los calorímetros electromagnético y hadrónico para medidas de energía. Finalmente se encuentra el espectrómetro de muones, que mide el momento de los muones y da el trigger.

- CMS (Compact Muon Solenoid) [6]: Es otro detector de propósito general que posee en mismo potencial que ATLAS aunque un diseño distinto.

- ALICE (A Large Ion Collider Experiment) [7]: Este detector está dedicado al estudio colisiones de iones pesados y del plasma quark-gluon.

- LHCb (Large Hadron Collider beauty [8]: Este detector está diseñado para estudiar violación de *CP* en interacciones de hadrones B. El LHCb es un espectrómetro lineal dispuesto en la misma dirección que el haz.

- TOTEM (Total Cross Section, Elastic Scattering and Diffraction Dissociation (TOTEM) [9]: El objetivo de este experimento es la medición de secciones eficaces, procesos difractivos y dispersiones elásticas.

- LHCf (Large Hadron Collider forward) [10]: Es un experimento que utiliza partículas producidas en las regiones hacia delante de las colisiones para simular de rayos cósmicos en condiciones de laboratorio.

Está previsto realizar una actualización del LHC para aumentar la luminosidad

instantánea en un factor diez, mejorando el potencial del acelerador para el estudio de la Física de Parículas [11]. Esta actualización se conoce como HL-LHC (High Luminosity-LHC). Está actualización se acometería en tres fases [12] en las que se iría aumentando la luminosidad instantánea hasta 10^{35} $cm^{-2}s^{-1}$ y la energía del haz hasta 14 TeV. La luminosidad integrada en el HL-LHC para cada experimento sería 3000 fb^{-1}. El HL-LHC presenta desafíos experimentales como el daño por radiación de los detectores o un aumento del fenómeno de apilamiento. Los detectores de trazas de ATLAS y CMS necesitarán ser reemplazados (detectores y electrónica). En concreto, los detectores de silicio sufrirán importantes daños por radiación y deberán ser completamente reemplazados.

Los detectores de silicio son ampliamente utilizados en los detectores de LHC para la reconstrucción de trazas de partículas con carga con gran precisión. Los requerimentos del LHC han llevado al límite de la tecnología a estos dispositivos. En futuros colisionadores de alta luminosidad, como el HL-LHC, serán necesarios detectores con propiedades mejoradas, como una alta resistencia a la radiación, una recolección de carga rápida y eficiente así como un espesor limitado.

La colaboración RD50 del CERN [13] intenta a dar respuesta a esta necesidad. Se dedica a la investigación y desarrollo de la tecnología adecuada de detectores para operar en futuros aceleradores de alta luminosidad. Las diferentes lineas de investigación de la colaboración pueden verse en la figura 1.5. El marco de trabajo principal de esta tesis es la línea de investigación *Full Detector Systems* de RD50. En esta linea de investigación se fraguó en 2005 una colaboración entre la Universidad de Liverpool, el Centro Nacional de Microelectrónica (CNM) y el Instituto de Física Corpuscular (IFIC), cuyo principal objetivo es el diseño, desarrollo e implementación de un sistema de lectura para detectores de microbandas de silicio. Esta colaboración se denomina ALIBAVA. En esta tesis se detalla el diseño, desarrollo e implementación de este sistema así como su actualización para toma de datos en *test beam*.

1.2. Detectores de silicio

Los detectores de silicio están basados en estructuras semiconductoras *pn* asimétricas. Normalmente, la unión se forma por difusión o implantación iónica de una delgada capa de tipo p^+ altamente dopada en un substrato débilmente dopado de tipo *n* (detector p^+-*n*). Otras posibilidades existen, como detectores n^+-*p* y n^+-*n*. La zona de desertización se extiende predominantemente en la región débilmente dopada, que se utiliza como zona activa de detección. Bajo el substrato se utiliza una capa altamente dopada del mismo tipo que éste para formar un contacto resistivo. Sobre los implantes altamente dopados se sitúan electrodos de aluminio.

En la figura 2.4. puede verse una vista esquemática de este tipo de detectores y su principio de funcionamiento. Se aplica al detector una tensión para polarizar inversamente las uniones *pn.* Esta polarización en inversa produce una zona de desertización en el detector libre de portadores de carga con un campo eléctrico. Al valor de la tensión de polarización para el que todo el volumen del detector está desertizada se denomina tensión de desertización total.

Cuando una partícula cargada cruza la zona desertizada, ésta pierde energía en procesos de ionización dando como resultado la generación de portadores de carga (pares electrón-hueco). La pérdida de energía de una partícula a través de un material es un proceso estocástico y viene dada por la velocidad, la carga y la masa de la partícula, así como la densidad del material [18, 19]. Para un detector de silicio de 300 μm de espesor a 300 K, la carga depositada por un *mip* (*minimum ionising particle*) que atraviese el detector es 22500 pares electrón-hueco o 3.6 fC.

Los pares electrón-hueco generados realizan un movimiento de deriva hacia los electrodos debido a la influencia del campo eléctrico. El movimiento de deriva es en dirección contraria para electrones y huecos. La velocidad de deriva de los electrones es unas tres veces mayor que la velocidad de deriva de los huecos en silicio a 300 K. La velocidad de deriva depende del campo eléctrico y de la movilidad de los portadores de carga.

En consecuencia, se produce en los electrodos un pulso de corriente proporcional la carga depositada por la partícula. El pulso de corriente se forma cuando los portadores de carga inician su movimiento, al inducir cargas imagen en los electrodos de igual magnitud. Estas cargas inducidas pueden calcularse según el teorema de Ramo [20]. La forma del pulso de corriente en un electrodo determinado depende de la contribución de electrones y huecos (se mueven en direcciones opuestas pero al tener carga contraria inducen corriente del mismo signo). La duración del pulso de corriente viene determinado por el tiempo de recolección de carga (unas tres veces mayor para los huecos que para los electrones). El tiempo de recolección de carga puede disminuirse polarizando el detector con un valor de tensión superior a la tensión de desertización total, aunque puede darse un proceso de avalancha y dañarse el detector. La carga total inducida en un electrodo del detector será la suma de la corrientes inducidas al moverse electrones y huecos generados.

La principal aplicación de los detectores de silicio en Física de Partículas son como detectores de posición para reconstruir las trazas de partículas cargadas. Los electrodos del sensor pueden segmentarse para formar microbandas o píxeles. Para

realizar medidas de posición precisas, se requiere una alta segmentación. Esto se consigue dividiendo el implante superior en pequeñas regiones, en el caso de detectores de microbandas en tiras estrechas y en el caso de detectores de píxeles en rectángulos. La resolución espacial en un detector de microbandas de silicio depende fundamentalmente de la geometría (anchura de las microbandas y separación entre ellas), la anchura de difusión transversal de la nube de portadores de carga y el método de lectura (analógico o binario). Para separaciones entre microbandas típicas de 20-200 μm se pueden obtener resoluciones con lectura binaria de 6-60 μm. Utilizando lectura analógica y métodos de división de carga resistivos o capacitivos, se puede mejorar la resolución espacial.

La lectura del pulso de corriente de los detectores de silicio se realiza, en primera instancia, por medio de un amplificador dado que la señal es de pequeña amplitud. La señal típica para un *mip* que atraviese un detector p^+-n de 320 μm de espesor totalmente desertizado se muestra en la figura 2.6. La duración de la señal es sobre 0.1-30 ns para detectores con espesor de 10-300 μm y viene dada por tiempo de recolección de carga. La señal tiene un bajo tiempo de subida debido al proceso de difusión de los portadores de carga [24, 25].

El circuito equivalente de un detector de microbandas de silicio puede representarse por medio de una fuente de corriente en paralelo con un condensador, como se muestra en la figura 2.7. La fuente de corriente proporciona el pulso de corriente inducido por los portadores de carga generados en la zona desertizada del detector por partículas cargadas. El condensador representa la capacidad total del detector. Un condensador de desacoplo (integrado en el detector o externo a él) debe utilizarse para minimizar la corriente de fugas en la lectura de la señal del detector.

Existen tres configuraciones básicas para el amplificador de lectura de detectores de microbandas de silicio [17, 26, 27]. El amplificador sensible a carga integra activamente la corriente del detector utilizando la capacidad de realimentación del amplificador. Es la mejor configuración respecto al ruido y tiene una impedancia de entrada resistiva. El amplificador de tensión integra la corriente del detector en la propia capacidad del detector, amplificando la tensión en esa capacidad. Está configuración es interesante si la capacidad del detector es prácticamente constante, lo cual se da en pocas ocasiones. Esta configuración es además más ruidosa. El amplificador de corriente amplifica directamente la corriente del detector convirtiéndola en una tensión de salida proporcional. La mayor desventaja es su impedancia de entrada inductiva para frecuencias de entrada superiores a la frecuencia de corte del amplificador. Es más ruidoso que el amplificador sensible a carga.

La utilización de un conformador de la señal posterior al amplificador tiene dos objetivos: evitar el efecto de apilamiento de la señal y mejorar la relación señal ruido. Existen dos tipos de conformadores [17, 24]: los que no varían su respuesta en el tiempo y los que varían su respuesta en el tiempo en función de una señal externa (u otro parámetro). Los más comúnmente utilizados para detectores de microbandas de silicio son los invariantes de tipo CR-RC^n. Se caracterizan por utilizar una etapa de diferenciación o filtrado paso bajo (CR, tiempo de respuesta, τ_d) y varias (n) de integración o filtrado paso alto (RC, tiempo de respuesta τ_i).

El ruido en los detectores de silicio procede de las fluctuaciones en el número y la velocidad de los portadores de carga que contribuyen a la corriente de salida del detector. La electrónica de lectura del detector (amplificador y conformador) también introduce ruido. Como consecuencia la relación señal ruido se ve degradada.

1.3. Especificaciones y arquitectura del sistema

El objetivo principal de la colaboración ALIBAVA es el diseño de un sistema de adquisición para detectores de microbandas de silicio que pueda ser usado con diferentes *setups* de laboratorio para investigar parámetros importantes de este tipo de detectores. El sistema debe cumplir las siguientes especificaciones:

- El sistema debe ser compacto y portátil para poder ser utilizado fácilmente en cualquier laboratorio.

- El sistema contendrá dos chips de lectura específicos para la lectura de detectores de microbandas de silicio de forma analógica. El particular, el chip a utilizar será el chip Beetle [41]. Estos chips se utilizan en el experimento LHCb del LHC.

- El sistema se podrá utilizar con dos *setups* de laboratorio diferentes: una fuente radiactiva y un láser. En ambos caso el sistema procesará y generará las señales necesarias. En el caso de la fuente radioactiva, el sistema tendrá dos entradas de *trigger* para las señales analógicas procedentes de dos fotomultiplicadores. Adicionalmente tendrá una señal de entrada de *trigger* auxiliar para señales pulsadas positivas o negativas, en corriente o tensión. El sistema realizará adquisiciones de datos a partir de estas señales En el caso del láser, el sistema generará una señal de *trigger* sincronizada para disparar la fuente del láser a través de un generador de pulsos. El sistema será capaz de retardar esta señal para realizar adquisiciones sincronizadas.

- El sistema se conectará a un computador donde se encontrará el *software* de adquisición mediante USB (*Universal Serial Bus*). En el computador se procesarán y almacenarán los datos adquiridos.

- El sistema será controlado desde el computador mediante un software de adquisición en comunicación con una FPGA que interpretará y ejecutará las órdenes. El software será diseñado *ad hoc* para el sistema y funcionara en el sistema operativo Linux.

- El sistema tendrá su propio sistema de alimentación. A partir de una fuente de alimentación portátil (del tipo de las utilizadas en los ordenadores portátiles) el sistema generará los niveles de alimentación que requiera. La alimentación de los detectores de silicio se realizará independientemente.

El objetivo principal del sistema será el procesado y digitalización de las señales analógicas procedentes de todos los canales de los detectores así como el procesado y generación de las señales necesarias para los dos tipos de *setups* de laboratorio.

El sistema se ha divido en dos partes principales: una parte *hardware* y otra software (figura 3.5). La parte *hardware* está compuesta por una placa madre y otra placa hija. La placa madre se ha concebido para procesar los datos analógicos procedentes de los chips de lectura, procesar o generar las señales de *trigger* necesarias para los dos tipos de *setups*, controlar el sistema completo y comunicarse con el software mediante USB. Contendrá la FPGA y otro hardware necesario. La placa hija es una pequeña placa que concebida para contener dos chips de lectura Beetle y su electrónica asociada. Esta placa contendrá a los detectores de silicio, cuyos canales serán conectados a los chips mediante adaptadores (*pitch-adaptors*) diseñados a tal efecto. La razón principal para dividir el hardware en dos placas es prevenir al resto del hardware del entorno agresivo que sufrirán los detectores (radiación y bajas temperaturas). Ambas placas se comunicarán mediante cable plano.

La parte software controlará todo el sistema y procesará los datos adquiridos de los detectores por la parte hardware para almacenarlos en un formato adecuado para posterior análisis. El software realizará la configuración del sistema, su calibración así como la adquisición de datos. También implementará el interfaz gráfico de usuario y monitorizará el estado del sistema en diferentes situaciones. El software se diseñará en dos niveles y podrá funcionar en el sistema operativo Linux.

2. El sistema de adquisición

2.1. La placa hija

El diagrama de bloques de la placa hija se muestra en la figura 4.1. Los dos componentes principales son dos chips *Beetle* para la lectura de hasta 256 canales de detectores de microbandas de silicio. Las señales analógicas de salida de ambos chips se conectan a una etapa de amplificación diferencial y son enviadas a la placa madre en paralelo. Una señal digital con formato LVDS (*low voltage differential signaling*), denominada *DataValid,* se envía a la placa madre desde cada chip *Beetle* para anunciar que se van enviar nuevos datos. Las señales de digitales de control lento (SDA y SCL) son enviadas desde la placa madre utilizando un bus I2C (*inter-integrated circuit*). Las señales digitales de control rápido (*Clk*, *Reset*, *Trigger* y *Testpulse*), con formato LVDS, son también enviadas desde la placa madre. Todas estas señales de control son compartidas por ambos chips *Beetle*.

Se ha situado un termistor tipo NTC (*negative temperature coefficient*) lo más cercano posible a los detectores para tener una lectura de la temperatura en cada adquisición de datos de los detectores. La tensión de alimentación para la placa hija (5 V *dc*) se envía desde la placa madre. Esta tensión de alimentación se regula en placa hija mediante tras reguladores lineales de tipo LDO (*low drop out*) para obtener las tensiones de alimentación necesarias para los chips Beetle (2.5 V *dc*) y la etapa de amplificación (3 V *dc*). La conexión ente la placa hija y la placa madre se realiza por medio de conectores tipo IDC (*insulated displacement connector*) y cable plano. La tensión de alimentación para los detectores se suministra independientemente a través de un conector de alimentación *Lemo*. Esta tensión de alimentación se filtra en la propia placa hija mediante una etapa RC doble para minimizar el ruido en la misma.

El chip *Beetle* se ha desarrollado en el laboratorio de ASICs de la Universidad de *Heidelberg* para el experimento LHCb. La principales características del chip pueden encontrarse en [41, 44, 45]. El diagrama de bloques del chip se muestra en la figura 4.2. El chip consta de 128 canales de entrada. Cada canal esta compuesto por un amplificador sensible a carga, un conformador CR-RC y un amplificador de salida. La señal de entrada procedente del detector puede ser tanto un pulso de corriente positivo como negativo. La forma del pulso de salida de la etapa analógica puede ser configurado. En este sistema se trabaja con los parámetros nominales del chip. Cada canal dispone de un comparador que puede configurarse para discriminar los pulsos de salida de la etapa analógica.

La señal de salida de la etapa analógica, o la señal de salida del discriminador, es muestreada a la frecuencia de reloj del chip en una memoria analógica circular. En este sistema se trabaja con la señal analógica sin discriminar y con la frecuencia de reloj nominal de 40 MHz. La señal de reloj (*Clk*) se suministra externamente. La memoria analógica tiene 130 x 187 celdas con una latencia programable (distancia entre datos que se escriben y que se leen de la memoria) de hasta 160 intervalos de muestreo. En este sistema la latencia es la nominal de 128 intervalos. Los datos almacenados en la memoria son leídos mediante un amplificador sensible a carga para cada canal. La lectura de una columna determinada de la memoria (128 canales) se controla con la señal de tipo LVDS denominada *Trigger*. Los datos correspondientes a los 128 canales de entrada son multiplexados en una o cuatro señales de salida. Se utilizan amplificadores de corriente para estas señales de salida. En este sistema se utiliza una sola señal de salida con los 128 canales multiplexados.

El formato de la señal de salida puede verse en las figuras 4.4 y 4.5. La señal consta de un encabezado de 16 bits con información sobre el chip y el valor analógico de la tensión muestreada para cada uno de los 128 canales. Esta tensión será proporcional a la carga recolectada en cada canal. La anchura de cada canal o bit del encabezado es de 25 ns dado que el multiplexado se realiza a 40 MHz. Junto con la señal de salida se genera la *DataValid*. Todas las corrientes y tensiones para alimentar las diferentes etapas analógicas del chip se generan mediante DACs (*Digital to Analogue Converter*) de 8 bits configurables por control lento. En este sistema se utilizan los parámetros nominales de alimentación para estas etapas. El chip dispone de un sistema de calibración consistente en una etapa de inyección de carga en cada canal. La cantidad de carga a inyectar en cada canal se configura mediante el control lento y el disparo del pulso en los canales se realiza mediante la señal de control rápido *Testpulse*. El chip ha sido implementado en tecnología estándar CMOS de 250 nm y es resistente a la radiación para dosis de hasta 130 Mrad.

Las señal de salida de datos de cada chip *Beetle* es una señal diferencial en corriente. Los niveles de la señal vienen dados por la ecuación 4.1 para los datos analógicos y por la tabla 4.3. para el encabezado. El ancho de banda de la señal es de 125 MHz. Cada una de estas dos señales han sido amplificadas mediante sendos amplificadores diferenciales de tensión basados en el AD8132 [50]. El esquema de este circuito se muestra en la figura 4.9. El amplificador tiene un ancho de banda de 160 MHz y ganancia dos. No se requiere realizar ecualización en estas señales si la longitud de cable plano es menor de 5 m.

Para conectar los canales de salida de los detectores a los canales de entrada de los dos chips *Beetle* se han utilizado adaptadores (*pitch-adaptors*) diseñados *ad hoc*. Estos adaptadores consisten en pistas y huellas de aluminio sobre un substrato de vidrio. Las conexiones entre los adaptadores y los chips, así como entre los adaptadores y los detectores, se realizan mediante pequeños hilos de aluminio unidos por microsoldadura por ultrasonidos. Los adaptadores se pegan a la placa mediante pegamento no conductivo. Los adaptadores han sido diseñados en el CNM y se muestran en la figura 4.15. Hay tres adaptadores uno para el chip (*Fan_chip*) y otro intermedio (*Fan_int*) pegados en la placa hija y otro (*Fan_det*) que va en otra pequeña PCB (*Printed Circuit Board*) adyacente a la placa hija donde se pegan los detectores. El objetivo es poder reutilizar una placa hija con diferentes detectores.

La placa hija ha sido diseñada en la Univesidad de Liverpool. Es una PCB de 72 mm x 76.2 mm que consta de cuatro capas (señal-masa-alimentación-señal) con componentes sólo en la capa superior (figura 4.17). También se han diseñado dos tipos diferentes de placas para los detectores (figura 4.19). Las dos placas se apoyan en un soporte diseñado a tal efecto (figura 4.20).

2.2. La placa madre

2.2.1. Diagrama de bloques y *hardware* de la placa hija

El diagrama de bloques de la placa madre se muestra en la figura 5.1. El bloque principal es una FPGA donde se ha implementado gran parte de la lógica digital que permite la funcionalidad del sistema controlando el resto de bloques. Desde un principio se ha restringido la implementación de la lógica digital diseñada a una FPGA debido a su versatilidad.

Las dos señales analógicas multiplexadas que proceden de los chips *Beetle* son llevadas en primer lugar a un bloque denominado *Signal Conditioning* (figura 5.2). En primer lugar cada señal de entrada analógica, señal diferencial en tensión, se transforma a una señal *single-ended* en tensión mediante un AD8130 [55]. Cada señal *single-ended* se envía a través de un amplificador LMH6559 [56] a un conector de salida *Lemo*. También se vuelve a realizar la conversión de cada señal *single-ended* a diferencial en tensión, mediante un AD8139 [57] para poder ser conectada a un ADC (*Analogue to Digital Converter*). En esta última conversión también se le suma a cada señal un nivel de tensión continua para que pasen de ser bipolares a unipolares y estén en el rango de conversión del ADC. Cada uno de los dos ADC, MAX1448 [58], se encarga de digitalizar la correspondiente señal analógica procedente de la placa hija. La resolución de los ADC es de 2 mV y la

frecuencia de muestreo es de 40 MHz. El rango de conversión de cada ADC es de ±1024mV sobre el nivel de tensión continua sumado en la etapa de acondicionamiento anterior. Los dos ADC se controlan mediante lógica digital implementada en la FPGA. Para ello son necesarias las dos señales de entrada (*DataValid0* y *DataValid1*) generadas por los chips *Beetle* al enviar datos en las señales analógicas multiplexadas. A partir del flanco de subida de cada una de estas señales digitales el bloque de control de cada ADC generará una señal de muestreo de 40 MHz para cada ADC, sincronizada de tal forma que se muestreará el punto medio de cada uno de los 128 canales analógicos correspondientes a cada señal analógica multiplexada, con una resolución temporal mínima de ±5 ns respecto al punto medio de cada canal.

Las señales digitales de control rápido (*Reset, Trigger*, *Clk* y *Testpulse*) así como las señales digitales de control lento (SDA y SCL) para los chips *Beetle* son generadas por sendos bloques de lógica digital diseñados a tal efecto. Las señales de control rápido son generadas con formato LVCMOS (*Low Voltage Complementary Metal Oxide Semiconductor*) a 3.3 V. Posteriormente se utilizan dos convertidores DS90LV047A [61] para pasar de formato LVCMOS a formato LVDS. Las señales *DataValid0* y *DataValid1* son enviadas desde la placa hija en formato LVDS y se utiliza un receptor DS90LV048A [64] para convertir su formato a LVCMOS. El uso de los convertidores se justifica en la protección de la FPGA. Para todas estas señales se utilizan choques de supresión de ruido en modo común con el fin de minimizar el ruido (figura 5.4). Para las señales de control lento se ha diseñado un bloque de comunicación I2C integrado en la FPGA, que es capaz tanto de escribir datos en los chips *Beetle* como de leer los datos escritos para comprobar si ha habido algún error en el proceso de escritura (figura 5.5).

Los bloques *Trigger Conditioning*, *TDC* así como parte de la lógica digital implementada en la FPGA se han diseñado para su uso con el *setup* de la fuente radioactiva. El bloque *Trigger Conditioning* está formado por dos discriminadores para las señales de *trigger* de entrada procedentes de dos fotomultiplicadores (TRIG IN1 y TRIG IN2) y por un conversor de nivel para una entrada bipolar pulsada (TRIG PULSE IN) en tensión (rango de ±5 V) o corriente sobre 50 Ω (rango de ±100 mA). Estos circuitos (figura 5.10) se han implementado utilizando dos comparadores MAX9601 [66]. Asimismo, se ha incluido un DAC de 12 bits, DAC7614 [70], para generar los cuatro niveles de tensión continua programables, utilizados en los discriminadores y el conversor de nivel (uno para cada discriminador y dos para el conversor de nivel). El rango dinámico de cada nivel de tensión está comprendido entre ±2048 mV y la resolución del DAC es de 5 mV. La finalidad de estos bloques es generar tres señales digitales pulsadas que estén activas cuando se supere el nivel de discriminación fijado (en el caso de TRIG IN1

y TRIG IN2) o cuando haya pulso positivo o negativo por encima de los umbrales fijados en conversor de nivel (TRIG PULSE IN). A partir de estas tres señales digitales pulsadas se generará una señal digital pulsada de *trigger* (TRIG IN), en función de la combinación de señales de *trigger* elegida por el usuario (TRIG IN1 o TRIG IN2, coincidencia de TRIG IN1 y TRIG IN2 o utilización solo de TRIG PULSE IN), mediante un bloque de lógica digital implementado en la FPGA. Además, el DAC es controlado por otro bloque de lógica digital desde la FPGA para que el usuario pueda programar los niveles de tensión utilizados en los discriminadores y el conversor de nivel.

El bloque *TDC* está formado por un TDC externo a la FPGA, TDC-GP1 [71]. No obstante, en función de la FPGA utilizada este bloque puede ser integrado en la misma como un bloque más de lógica digital. El TDC se encarga de realizar una medida de tiempo entre el flanco de subida de la señal digital pulsada de *trigger* (TRIG) generada en la FPGA a partir de las señales de entrada de *trigger* (en función de la combinación elegida por el usuario) y el flanco de subida de una señal digital periódica de referencia, a partir de la cual se dará orden a los chips *Beetle* (mediante la activación de la señal TRIG_R que a su vez activa la señal de control rápido *Trigger* tras la latencia de 3.2 μs) de que una determinada muestra de la señal adquirida continuamente por los chips *Beetle* quiere ser leída. La señal periódica de referencia es una señal de periodo 100 ns y anchura de pulso de 25 ns. El TDC mide entonces el tiempo transcurrido entre el flanco de subida de una señal de activación (TRIG) y el flanco de subida de una señal de parada (pulso correspondiente a la señal de referencia periódica), la cual sólo será enviada al TDC si anteriormente se ha enviado la señal de activación. Todo el control del TDC y la generación de las señales de activación y parada son realizados por un bloque de lógica digital implementado en la FPGA. De esta manera se puede llegar a reconstruir el pulso analógico muestreado en los chips Beetle (figura 4.3) tomando múltiples adquisiciones a partir de señales activas de *trigger* de entrada aleatorias en el tiempo (como es el caso de la generación de partículas en una fuente radioactiva). Para que esta reconstrucción sea lo suficientemente precisa tiene una resolución en el TDC mayor de 1 ns y un rango de medida 100 ns.

El bloque *Programmable Delay*, *50 Ω Driver* así como parte de la lógica digital implementada en la FPGA se han diseñado para su uso con el *setup* del láser. El bloque *Programmable Delay* (figura 5.12) consta de un circuito digital de retraso programable, 3D7428-1 [71]. Este circuito es capaz de retrasar pequeños intervalos de tiempo, 1 ns, la señal pulsada digital de salida de *trigger* (TRIG OUT) en función del retraso programado. El bloque *50 Ω Driver* (figura 5.12) permite que la señal TRIG OUT pueda ser conectada directamente a cualquier entrada de 50 Ω sin distorsionarla. El circuito digital de retraso se controla desde un bloque de lógica

digital implementado en la FPGA. Este mismo bloque es el que genera la señal TRIG OUT. En principio esta señal debe tener un formato de salida LVCMOS (3.3 V). El bloque de lógica digital también genera una señal digital (TRIG_L) a partir de la cual se dará orden a los chips Beetle (mediante la activación de la señal de control rápido *Trigger* tras una latencia de 3.2 μs) de que una determinada muestra de la señal adquirida continuamente por los chips *Beetle* quiere ser leída. El retraso entre el flanco de subida de TRIG OUT (antes de pasar por el circuito digital de retraso del bloque *Programmable Delay*) y el flanco de subida de la señal de control rápido *Trigger* puede ser variado en pasos de 25 ns por el bloque de lógica digital. Así, conjugando ambos retrasos, se puede ir adquiriendo datos de los chips *Beetle* correspondientes a un determinado punto del pulso analógico (figura 4.1). Por tanto, el sistema es capaz de reconstruir el citado pulso tomando adquisiciones correspondientes a diferentes puntos muestreados del pulso analógico a partir de la señal sincronizada de TRIG OUT. El usuario será el que programe el rango de retraso total y el intervalo mínimo de retraso entre diferentes muestras y el sistema sincronizará automáticamente las señales TRIG OUT y *Trigger* para realizar las adquisiciones con los parámetros de retraso programados.

El bloque *Digital Converter* está formado por un conversor analógico-digital específico, MAX6682 [75], para digitalizar la señal procedente del termistor, *i.e.* una resistencia de tipo NTC. De esta forma, se pueden adquirir datos de temperatura además de los datos procedentes de los chips *Beetle*. El control del conversor se realiza desde un bloque de lógica digital implementado en la FPGA, que se encarga de la lectura de los datos digitalizados por el conversor. La digitalización de los datos se realiza automáticamente con una frecuencia de 1 Hz. La comunicación entre el conversor y la lógica digital se realiza siguiendo un formato de comunicación serie, puesto que la frecuencia de conversión es baja.

El bloque *SDRAM* está constituido por una memoria SDRAM de 256 Mb, MT48LC32M8A2-7E [77], externa a la FPGA con el fin de almacenar los datos adquiridos de los chips *Beetle* y los datos de temperatura. Además, los datos obtenidos mediante el TDC en el caso del *setup* de la fuente radioactiva también son almacenados en esta memoria junto con los demás datos. Se ha elegido una memoria SDRAM debido a que las memorias de este tipo son las que mayor capacidad de almacenamiento tienen (cientos de megabits por chip). De esta forma se podrán almacenar en la memoria los datos correspondientes a un número significativo de eventos (*i.e.* decenas de miles). Esta memoria es controlada mediante un bloque de lógica digital implementado en la FPGA, con el cual se puede leer y escribir datos en la memoria. Además, también genera las señales digitales necesarias para el refresco y el direccionamiento de la memoria.

El bloque *USB* (figura 5.17) está formado por un circuito integrado específico, FT245R [80], que gestiona la comunicación con el software siguiendo el protocolo de comunicación USB 2.0. Se ha diseñado un bloque de lógica digital implementado en la FPGA que controla este circuito integrado, pudiendo leer y escribir datos en el circuito controlador USB de forma sencilla. El circuito controlador USB se encarga de enviar los datos que le son escritos desde la FPGA en formato USB. Asimismo, se encarga de recibir los datos enviados en formato USB desde el software y de ponerlos a disposición del bloque de lógica digital que lo controla.

El bloque *FPGA* está compuesto de una FPGA, Spartan-3 XC3S400-PQ208 [89], en la que se implementa todos los bloques de lógica digital citados anteriormente. Asimismo, el bloque incluye un oscilador integrado que genera la señal de reloj utilizada por la FPGA, de frecuencia 40 MHz, así como un circuito generador de una señal de *reset* para la FPGA a partir de un pulsador. Por último, el bloque incluye los circuitos necesarios, una memoria PROM (*Programmable Read Only Memory*) y un conector específico, para la programación de la FPGA.

Para la alimentación de todo el *hardware* se ha implementado el bloque *Supply System*. La finalidad de este bloque es la generación de las tensiones de alimentación necesarias en la placa madre y la placa hija. La generación de estas tensiones se realiza a partir de una tensión continua de entrada de 5 V. Para ello se utilizan conversores DC-DC conmutados y aislados. De esta forma las tensiones de alimentación para los circuitos analógicos y digitales son generadas de forma independiente para que el ruido de las tensiones de alimentación digitales no afecte a las tensiones de alimentación analógicas. Asimismo, se incluyen filtros en la entrada y salida de cada conversor DC-DC para minimizar el ruido generado por este tipo de conversores. Además, se incluye también un filtro EMI (*Electromagnetic Interference*) integrado lo más próximo posible a la entrada de la tensión de alimentación principal (5V).

El *hardware* de la placa madre ha sido implementado en una PCB de 235 x 160 mm (figura 5.24). Debido a la complejidad del diseño la PCB consta de seis capas (figura 5.25). Las capas superior e inferior se han utilizado para las señales. En la inferior se han rutado solo señales digitales, algunas de impedancia controlada. En la superior señales analógicas, algunas con impedancia controlada, y digitales no ruidosas. La segunda capa (figura 5.28) se ha utilizado para planos de masa analógico y digital. La tercera capa (figura 5.31) se ha utilizado para planos de alimentación analógicos, digitales y para la placa hija. En la cuarta capa (figura 5.32) se tienen planos de alimentación digitales y de la placa hija. En la quinta capa (figura 5.29) se han incluido los planos de masa digital y de la placa hija. Los

componentes se han situado en la capa superior con especial cuidado para separar los circuitos analógicos, digitales y de potencia (figura 5.26). Todos los circuitos y planos de alimentación/masa correspondientes al sistema de alimentación están separados del resto de componentes para minimizar el ruido.

2.2.2. Lógica de la FPGA

En la figura 5.34 se muestra el diagrama de bloques conceptual de la lógica implementada en el interior de la FPGA. El bloque *CFSM* (*Central Finite State Machine*) se encarga de controlar todo el hardware del sistema a partir de las órdenes recibidas desde el software por USB. Para ello, en función del estado del sistema, el bloque CFSM se encarga de accionar diferentes bloques de lógica digital integrados en la FPGA.

El bloque *Beetle Slow Control* (figura 5.37) es el encargado de controlar el bus I2C mediante el cual se pueden configurar los registros internos de ambos chips *Beetle*. Este bloque es capaz de direccionar todos los registros de configuración de cada chip *Beetle*. Asimismo, es capaz de comprobar la consistencia de los datos escritos avisando si ha encontrado algún error.

El bloque *Beetle Fast Control* (figura 5.36) se encarga de generar las señales de salida de control rápido (*Reset*, *Trigger, Clk* y *Testpulse*) para los chips *Beetle* en función del estado del sistema. La señal *Testpulse* se genera a partir de una señal interna de calibración (CAL, generada desde la CFSM) para inyectar una determinada carga (cuyo valor se configura mediante el control lento) en cada canal de entrada del chip. A partir de la activación de *Testpulse*, se activará automáticamente (tras una latencia de 3.2 μs) la señal de *Trigger* para la adquisición de los datos muestreados correspondientes a ese pulso de calibración. Asimismo, la señal de *Trigger* se activará a partir de la señal TRIG_L (en el caso del *setup* láser) o de la señal TRIG_R (en el caso del setup de la fuente radioactiva) teniendo en cuenta la latencia de los chips Beetle (3.2 μs) y el retardo interno en la FPGA. También se activará la señal de *Trigger* a partir de la señal interna PD_TRIGGER (controlada desde la CFSM) con el fin de adquirir muestras de los chips *Beetle* para estimar el nivel correspondiente de carga recolectada nula de las señales analógicas de los chips *Beetle*.

El bloque *Trigger Out* (figura 5.43) se utiliza con el *setup* láser. Se encarga de generar la señal de *trigger* que excita el láser (TRIG OUT) y una señal interna (TRIG_L) que se envía al bloque *Beetle Fast Control*. Este bloque controla el circuito digital de retraso programable que se encarga de retrasar la señal TRIG OUT en intervalos 1 ns. Asimismo, el bloque es capaz de retrasar la generación de

un pulso en TRIG_L respecto a la generación de un pulso en TRIG OUT en intervalos de 25 ns. Combinando adecuadamente estos dos retrasos se consigue adquirir los datos correspondientes a la muestra de un determinado punto del pulso analógico generado en los chips *Beetle*. El bloque puede ir generando pulsos con retrasos diferentes o bien ser programado un determinado retraso fijo y que el bloque genere TRIG OUT y TRIG_L con una frecuencia fijada en 1 kHz.

El bloque *Trigger In* (figura 5.38 y 5.39) se utiliza con el *setup* de la fuente radioactiva. El bloque incluye la lógica digital de coincidencia, multiplexado y generación de pulsos necesaria para generar una señal pulsada de *trigger* (TRIG IN) que se conecta directamente al bloque de lógica digital *TDC Control*. Las señales de entrada a este bloque son las señales de salida de los discriminadores (SIN1 y SIN2) y del conversor de nivel (PPOS y PNEG) que se han integrado en el bloque *hardware* de denominado *Trigger Conditioning*. El bloque es programable desde la *CFSM* con el fin de que el usuario seleccione la coincidencia de SIN1 y SIN2, la utilización SIN1 o SIN2, o la utilización de PNEG o PPOS para la generación del pulso en TRIG. El bloque *DAC Control* (figura 5.41) incluye el control del circuito de DAC interno con el que se generan los cuatros niveles de tensión continua utilizados en los discriminadores y conversor de nivel del bloque *Trigger Conditioning*.

El bloque *TDC Control* (figura 5.42) integra la lógica digital necesaria, para realizar el control del circuito TDC externo para su configuración así como la lectura de los datos correspondientes a cada medición del TDC. Este bloque también se encarga de generar las señales de activación y parada del circuito TDC con las especificaciones descritas en la sección 2.2.1. Asimismo, este bloque generará la señal pulsada TRIG_R, conectada directamente al bloque *Fast Control*, a partir de la señal pulsada de parada del TDC con el fin de dar orden a los chips *Beetle* para proceder a la lectura de los datos correspondientes.

Cada bloque *ADC Control* (figura 5.35) controla uno de los dos circuitos ADC en este sistema. De esta manera, cada bloque *ADC Control* se encarga de generar de manera sincronizada la señal de muestreo (SCLK) de frecuencia 40 MHZ para cada circuito ADC. La señal de muestreo estará activa sólo durante el muestreo de los 128 canales multiplexados de la correspondiente señal analógica de cada uno de los chips *Beetle*. La sincronización se realiza a partir del flanco de subida de la correspondiente señal *DataValid* procedente de cada uno de los chips *Beetle*, con el fin de muestrear el centro (con una resolución mínima de ±5 ns) de cada uno de los 128 canales multiplexados. Asimismo, cada bloque debe ser capaz de leer los datos digitalizados procedentes de cada circuito ADC y de almacenarlos en una memoria FIFO (*First In First Out*) incluida en la FPGA.

El bloque *USB Control* (figura 5.44) incluye el interfaz necesario para la comunicación en modo paralelo con el circuito USB externo (lectura de datos recibidos desde el software y escritura de datos para ser enviados al software) y su control.Para ello, se incluyen dos memorias FIFO, una de lectura y otra de escritura. El bloque *Temperature Control* (figura 5.45) se encarga de controlar el conversor ADC externo que digitaliza la señal de temperatura procedente de la placa hija. El bloque *LED Control* (figura 5.46) se ha diseñado para el accionamiento de los dos LED de la placa madre.

El bloque *Clock Generator* (figura 5.50) se ha diseñado para generar todas las señales de reloj necesarias para el funcionamiento de la lógica secuencial de los bloques de la FPGA a partir de la señal de reloj de 40 MHz exterior. Todos los bloques de la FPGA funcionarán con una señal de reloj común de 40 MHz de frecuencia. Además, se generan otras tres señales de reloj desfasadas 90º (6.25 ns), 180º (12.5 ns) y 270º (18.75 ns) respecto a la señal de reloj común a todos los bloques. También, se generará una señal de 200 MHz sincronizada con la señal de reloj de 40 MHz común a todos los bloques. Estas señales de reloj son utilizadas por cada uno de los bloques ADC Control para generar la señal de muestro correspondiente sincronizada respecto a la correspondiente señal de *DataValid*. Por otra parte, este bloque también genera una señal de *reset* asíncrona a partir de la señal de *reset* externa a la FPGA. Esta señal de *reset* se envía a la CFSM que generará las señales de *reset* correspondientes para cada bloque de la FPGA.

El bloque *CFSM* y el bloque *SDRAM Control* han sido diseñados como un sistema embebido compuesto por un microprocesador software (figura 5.47) y un periférico prediseñado para el control de la memoria SDRAM externa (figura 5.48). La comunicación entre el sistema embebido y el resto de bloques de la FPGA se realiza mediante seis memorias FIFO y bloque de registros denominado *Arbiter*. En la figura 5.49 se muestra el diagrama de bloques que incluye el sistema embebido, el bloque *Arbiter* y el resto de bloques ya descritos. Dos memorias FIFO (de 128 por 32 bits) se utilizan para comunicación directa entre el microprocesador y los dos bloques *ADC Control*. Otras dos memorias FIFO (16 por 32 bits) se utilizan para la comunicación directa entre el microprocesador y el bloque *USB Control*. Las otras dos memorias FIFO (16 por 32 bits) se comunican con el bloque *Arbiter* y son utilizadas para la comunicación general entre los bloques y el microprocesador en ambos sentidos. El bloque *Arbiter* se encarga de gestionar esta comunicación mediante una FSM y un bloque de registros (tabla 5.6) en función de los datos escritos o demandados por el microprocesador.

Mediante el uso de un sistema embebido con microprocesador se gana

flexibilidad, puesto que la funcionalidad del sistema se programa mediante un lenguaje de alto nivel como un *firmware* que se ejecuta en el microprocesador. De esta forma, se pueden realizar cambios en el diseño del sistema modificando el *firmware* pero sin cambiar el diseño de los bloques. En la tabla 5.7 se muestran los recursos de la FPGA utilizados por este diseño.

2.2.3. *Firmware* de la FPGA

La funcionalidad del sistema viene determinada por el *firmware* del microprocesador, diseñado como una FSM que contiene todos los posibles estados por los que tiene que pasar el sistema para realizar lo que le corresponde. Los estados implementados en la CFSM se pueden ver en la figura 5.53.

En el estado *Reset* (tabla 5.8) se inicializa todo el *hardware* del sistema así como los bloques de lógica digital de la FPGA. Después de la inicialización, se comprueba la comunicación USB con el software mediante el intercambio de una serie de códigos. Si el sistema se ha inicializado correctamente y la comprobación de la comunicación USB es correcta, se va directamente al estado *Waiting*. En el estado *Waiting* el sistema espera a recibir desde el *software* la orden para ir a cualquier otro estado. El *hardware* al entrar en cada estado envía al *software* un código para hacerle saber en el estado que se encuentra (tabla 5.9).

En el estado *Beetle Configuration* (tabla 5.10) se procede a la configuración de los parámetros de funcionamiento chips *Beetle*. El *software* envía la dirección de cada registro y el valor a escribir en el mismo. Se escribe y se comprueba lo escrito. Si existe algún error se envía un código al *software* informando de ello y se activa el LED rojo. Una vez que se han configurado los registros internos de cada chip *Beetle* correctamente el *software* envía un código para que se vuelva al estado *Waiting*. En el estado *Trigger In Configuration* (tabla 5.11), se procede a la configuración del esquema de entradas de *trigger* a utilizar y de los niveles de tensión del DAC. Una vez que se han configurado estos dos bloques el software envía un código para que se vuelva al estado *Waiting*.

En el estado *Calibration* (tabla 5.12) se realizan adquisiciones de datos de calibración de los chips *Beetle*. En primer lugar se configura mediante el bloque *Slow Control* la amplitud del pulso a inyectar (en electrones) en los canales de entrada de cada chip *Beetle* a partir del dato enviado desde el *software*. Posteriormente, se activa la señal de calibración (CAL) del bloque *Fast Control*. De esta forma, se procede configurando distintas amplitudes del pulso a inyectar para obtener los datos de calibración correspondientes a cada uno de los 256 canales de los chips *Beetle*. Cuando se han obtenido los datos de calibración

requeridos por el usuario, el *software* envía un código para que se vuelva al estado *Waiting*.

En el estado *Laser Synchronization* (tabla 5.13), se realizan adquisiciones utilizando el sistema láser, variando el desfase en pequeños intervalos entre la señal que excita el láser, y la señal de control rápido *Trigger*. El rango de variación de este retraso y el tamaño de los intervalos son configurables por el usuario. De esta forma se realizan un número determinado de adquisiciones (configurable por el usuario) para cada intervalo de tiempo programado hasta que se ha recorrido el rango de retraso programado. Por tanto, se puede reconstruir a partir de estos datos el pulso analógico de los chips *Beetle* así como conocer el retraso a programar para muestrear un determinado punto del pulso. Los datos de cada adquisición (el dato de amplitud digitalizado correspondiente a cada uno de los 256 canales de entrada de los chips *Beetle*) son enviados directamente al *software* una vez adquiridos. Para cada adquisición se comprueba si ha ocurrido algún error en la digitalización de los datos adquiridos enviando en tal caso un código de error al *software*. Cuando se han adquirido los datos programados el *software* envía un código para que vuelva al estado *Waiting*.

En los estados *Laser Acquisition, RS Acquisition* y *Pedestals Acquisition* se realizan adquisiciones utilizando el sistema láser, la fuente radiactiva o sin señal. En *Laser Acquisition* (tabla 5.14) el usuario puede configurar un retraso fijo (en ns) entre las señales TRIG OUT y *Trigger*, para realizar la adquisición de un número programable de eventos. Para cada pulso de TRIG OUT se digitalizarán los datos correspondientes a los chips *Beetle* y un dato de temperatura. En *RS Acquisition* (tabla 5.15) el usuario configura el número de eventos que deben adquirirse. Para cada pulso de TRIG_IN se digitalizarán los datos correspondientes a los chips *Beetle*, un dato de temperatura y el dato de tiempo medido con el TDC. En *Pedestals Acquisition* (tabla 5.16), tras configurar el número de eventos a adquirir, para cada pulso de PD_TRIGGER se digitalizarán los datos correspondientes a los chips *Beetle* y un dato de temperatura. En estos estados, los datos adquiridos se van almacenando en la memoria SDRAM de la placa madre para aumentar la rapidez del sistema. Asimismo, para cada evento se comprueba si ha ocurrido algún error en la digitalización de los datos adquiridos o en su almacenamiento enviando en tal caso un determinado código de error al *software*. Cuando se han adquirido los datos programados el *software* envía un código para que vuelva al estado *Waiting*.

En los estados *Laser Reading* (tabla 5.17), *RS Reading* (tabla 5.18) y *Pedestals Reading* (tabla 5.19), se leen los datos almacenados en la memoria SDRAM de la placa madre la última vez que se accedió al correspondiente estado de adquisición (*Laser Acquisition*, *RS Acquisition* y *Pedestals Acquisition*, respectivamente). En

estos estados la placa madre envía al software el dato de número de eventos almacenados y espera a recibir un código del *software* para empezar a enviar los datos almacenados en la SDRAM. Cuando se recibe el código, se envían todos los datos almacenados y se regresa automáticamente al estado *Waiting*. Si hay algún error de lectura de datos de la SDRAM se envía al *software* el código de error correspondiente.

2.3. El *software*

La función principal del *software* es la de controlar el sistema y procesar los datos adquiridos por el *hardware*, así como almacenar los datos una vez procesados en un formato adecuado para que puedan ser analizados utilizando otros tipos de *software* para el análisis de datos en física de partículas.

El *software* controla el *hardware* del sistema a partir de códigos enviados por USB y que son interpretados por la lógica implementada en la FPGA de la placa madre. Además, la placa madre también envía al *software* los datos adquiridos por el *hardware* y códigos informando sobre el estado del *hardware*. De esta manera, el *software* puede controlar el *hardware* para su configuración, calibración y adquisición de datos con los dos tipos de *setup* (láser y fuente radioactiva). Por otra parte, el *software* procesa los datos adquiridos (datos de temperatura, medidas de tiempo con el TDC y datos digitalizados procedentes de los chips *Beetle*), para que tengan significado físico. Los datos introducidos por el usuario para configurar el sistema también son procesados por el *software* para que puedan ser entendidos por la placa madre. Este procesamiento a bajo nivel se lleva a cabo por el *software* en vez de por la lógica digital de la FPGA para aprovechar la capacidad de procesamiento del computador, dejando la capacidad de la FPGA para el control del *hardware* y la adquisición de datos en tiempo real.

Por otra parte, el *software* incluye un interfaz entre el usuario y el *hardware* para configurar y controlar el sistema de acuerdo con las preferencias del usuario así como para monitorizar el estado del sistema y los datos adquiridos para informar al usuario. El interfaz ha sido implementado como un interfaz gráfico de usuario (GUI). Finalmente, el *software* almacena los datos adquiridos y procesados a bajo nivel generando un archivo de salida compatible con otros tipos de *software* de análisis de datos utilizados en física de partículas, como ROOT [109].

Teniendo en cuenta lo expuesto anteriormente, el software se ha implementado en dos niveles conceptuales (figura 6.1): una parte de bajo nivel (l*ow level software*) y una parte de alto nivel (*high level software*). La parte de bajo nivel incluye los módulos de comunicación por USB y de procesamiento de datos a bajo

nivel. La parte de alto nivel incluye el interfaz gráfico de usuario para la comunicación entre el usuario y el sistema, la monitorización de los datos así como la generación de los archivos de salida.

El *software* ha sido diseñado para ser compatible con el sistema operativo Linux. Tanto la parte de bajo nivel como la de alto nivel se han implementado en un mismo paquete de *software* y se han diseñado utilizando el lenguaje de programación C++ [110]. Una descripción más detallada del software puede encontrarse en [111].

3. Desarrollo, producción y mejora del sistema

3.1. Desarrollo del sistema

Se desarrolló un prototipo de la parte *hardware* del sistema para verificar su funcionalidad así como para desarrollar y testar la parte software. Para los primeros tests con el prototipo se desarrollaron algoritmos en el entorno MATLAB [112] que se comportaban de manera similar al *software*. Como consecuencia de los test realizados con este prototipo se realizaron algunos cambios para mejorar la funcionalidad del sistema.

En primer lugar, se mejoró el diseño del bloque de lógica de la FPGA *Beetle Slow Control* para que, además de escribir en los registros de configuración de los chips, fuese capaz de leer lo escrito y comprobar su corrección.

En segundo lugar, se mejoró la sincronización de la señal de muestreo de cada ADC respecto a la señal de datos multiplexada procedente de los chips Beetle para evitar variaciones en la ganancia (figura 7.1). Para ello se cambió el diseño de cada bloque *ADC Control* para que la señal de muestreo del ADC digitalize justo el punto medio de cada canal de la señal multiplexada (figura 7.2) con una resolución de ±3.125 ns.

En tercer lugar, se implementaron los estados *Pedestals Acquisition* y *Pedestals Reading* en el *firmware* de la FPGA para poder adquirir los pedestales, *i.e.* señal para carga cero en la entrada de cada canal de los chips Beetle.

Finalmente, se dobló la ganancia del sistema para tener mayor resolución en medidas con la fuente radiactiva. Para ello se redujo el fondo de escala de los ADCs de ±1024 mV a ±512 mV, aumentando su resolución de 2 mV/ADC a 1

mV/ADC. Esta solución minimiza el ruido respecto a otras opciones. Tras estas mejoras se solucionaron las desviaciones en la ganancia observadas en la figura 7.1, como puede verse en la figura 7.3.

3.2. Medidas con sensores irradiados y no irradiados

En primer lugar, se realizaron medidas con un detector *p*+-*n* y otro detector *n*+-*p*, ambos sin irradiar. Ambos detectores tenían 128 canales, un espesor de 300 μm y una separación entre microbandas de 80 μm. La longitud de las microbandas era de 1 cm. Los detectores se conectaron a un chip Beetle de sendas placas hijas (canales 129-256). Los otros chips Beetle de ambas placas se dejaron sin conectar (canales 1-128). Se realizaron medidas con un láser y con una fuente β. Previamente se tomaron datos de calibración y pedestales para cada placa hija. Todas las medidas se realizaron a 20 ºC y con los detectores totalmente desertizados.

Las medidas de calibración corresponden a carga inyectada a la entrada de cada canal de los chips Beetle. Se inyectaron cargas desde 0 a 102500 electrones en pasos de 1025 electrones. El signo de la carga inyectada alterna para cada canal. Se adquirieron 100 muestras para cada paso. A partir de estos datos se obtuvo la curva de ganancia del sistema y se calculó la ganancia (cuentas de ADC/electrón) para cada canal de entrada de los chips (figura 7.6 para el detector *n*+-*p* y figura 7.7 para el detector *p*+-*n*). Se puede observar que la ganancia es diferente para los canales conectados a los detectores respecto de los que no lo están debido a la capacidad del detector. La ganancia no varía de forma significativa para los canales conectados al detector.

Se realizaron medidas con un láser con longitud de onda de 1060 nm (infrarrojo cercano) y una energía de los fotones de 1.17 eV. Se realizó una adquisición en modo *Laser Synchronization* con un rango de 100 ns en pasos de 1 ns. Para cada paso se tomaron 100 muestras. En las figuras 7.8 y 7.9 se muestra la carga recolectada (electrones) en función del retraso programado (ns) para el detector *p*+-*n* y el detector *n*+-*p* respectivamente. Se puede ver que se reconstruye el pulso analógico de los chips Beetle.

Se realizaron medidas con una fuente $^{90}Sr/^{90}Y$ β de baja actividad. Se usó como señal de *trigger* la señal de salida de un fotomultiplicador conectado a un centellador, que estaba situado bajo el detector y la fuente. La señal se conectó a la entrada TRIG IN1 y el nivel de discriminación se situó en -40 mV. Se adquirieron 20000 eventos con cada detector. En la figura 7.10 se representa la reconstrucción del pulso analógico del chip Beetle para el detector *p*+-*n*. El espectro de la señal

(número de eventos respecto al valor absoluto de la carga en electrones) con un corte entre 12 ns y 22 ns, se muestra en la figura 7.11 para el detector *n+-p* y en la figura 7.12 para el detector *p+-n*. Se puede ver en estas figuras que el espectro es consistente con una distribución de *Landau* en convolución con otra de *Gauss*. El pico de estos espectros corresponden con el valor más probable de un mip (*minimum ionizing particle*).

El ruido correspondiente a la placa hija con el detector *n+-p* conectado se representa en la figura 7.13 (ruido en electrones por canal). El ruido de los canales conectados al detector es mayor que el ruido de los canales no conectados. Teniendo en cuenta el valor de pico del espectro de la figura 7.11 (26940 electrones) y el valor del ruido en la figura 7.13 (690 electrones) se obtiene un SNR (*signal to noise ratio*) de 38 para el detector *n+-p.* Un resultado similar se obtendría para el detector *p+-n*.

En segundo lugar, se realizaron medidas en el IFIC con detectores irradiados. Las medidas realizadas con detectores de microbandas *n+-p* de 320 μm y 128 canales, tanto el láser infrarrojo como con la fuente $^{90}Sr/^{90}Y$ β, se reportan en [113]. El mismo tipo de medidas se realizaron con detectores de microbandas *n+-n* de 300 μm y 128 canales [114]. Se obtuvieron valores SNR mayores que 30 en estas medidas con sensores irradiados.

3.3. Producción del sistema

Se realizó una producción inicial de 50 placas hija y 5 placas madre para su uso en la colaboración ALIBAVA. Diferentes miembros de la colaboración RD50 mostraron su interés en utilizar el sistema, así que se realizó una nueva producción de 22 placas madre en noviembre de 2008. El interés mostrado por otros grupos de investigación ha causado la producción de más sistemas en 2009 y 2010, como se muestra en la tabla 7.1. Se han implementado diferentes controles de calidad durante el proceso de producción del sistema para asegurar su fiabilidad.

En total se han distribuido 34 sistemas, además de 35 placas hijas extra, a diferentes instituciones internacionales como puede verse en la tabla 7.2. En estas instituciones se han realizado medidas con diferentes tipos de sensores de microbandas irradiados [116-120]. También se han realizado medidas con sensores de silicio de doble cara de tipo 3D conectados en modo *dc* [121,122] así como con sensores de microbandas 2D [123].

3.4. Mejora del sistema

Se han implementado diversas mejoras en el sistema basadas en la experiencia propia al realizar medidas con el mismo así como en base a las sugerencias de otros usuarios del sistema.

En primer lugar, se ha aumentado la tensión de alimentación de los amplificadores de la placa hija de 3 V a 4.5 V cambiando un regulador de tensión en la misma. El objetivo era aumentar el límite superior del rango en modo común de la señal de entrada de 1 V a 2.5 V, con el fin de que al realizar medidas con la placa hija a bajas temperaturas (*i.e.* -30 ºC) la ganancia sea unas 1,4 veces superior que a temperatura ambiente (*i.e.* 20 ºC) y no 1,85 veces inferior como se había observado.

En segundo lugar, se ha mejorado el diseño de los *pitch-adaptors* con el fin de que la alimentación de los detectores pueda realizarse a través de los *pitch-adaptors* y no mediante dos cables independientes conectados entre la placa hija y la placa de los detectores. Para ello se ha incluido una pista adicional en los *pitch-adaptors* para la tensión de alimentación de los detectores (figura 7.23).

En tercer lugar, se ha modificado la lógica y el *firmware* de la FPGA para que el sistema pueda trabajar sólo con un chip Beetle operando. Para ello se han modificado los bloques de lógica *Beetle Slow Control* y *Arbiter* para que se pueda seleccionar con que chips se quiere trabajar. También se han modificado los estados del *firmware Beetle Configuration, Calibration* y *Acquisition.* El *software* también se ha modificado para incluir la selección de chip (figura 7.24).

En cuarto lugar, se se ha modificado la lógica y el *firmware* de la FPGA para que el sistema digitalice el encabezado de las señales multiplexadas procedentes de los chips Beetle además de los 128 canales analógicos. Para ello se han modificado los bloques de lógica *ADC Control* y *Arbiter* para que se puedan incluir los datos correspondientes a los 16 bits del encabezado. También se han modificado los estados del *firmware Calibration, Acquisition* y *Reading*. El *software* se ha modificado para mostrar y almacenar los datos digitalizados de los encabezados (figura 7.25).

En quinto lugar, se ha mejorado la adquisición de datos para calibración permitiendo que el retardo entre las señales de control rápido del chip Beetle *Testpulse* y *Trigger* pueda configurarse por el usuario en pasos de 1 ns y con un rango de 255 ns. De esta manera se puede reconstruir el pulso analógico característico del chip Beetle con datos de calibración (figura 7.27). Para ello se

han modificado los bloques de lógica *Beetle Fast Control* y *Arbiter* para que se pueda implementar el retardo programable. También se ha modificado el estado del *firmware Calibration*. El *software* se ha modificado para mostrar la reconstrucción del pulso en calibración y para configurar esta adquisición (figura 7.26).

Por último, se ha mejorado el reset por *software* del sistema y se ha modificado el *firmware* de la FPGA para que el *software* sepa con qué versión del *firmware* de la FPGA se está trabajando. Los recursos de la FPGA utilizados incluyendo estas mejoras se muestran en la tabla 7.3.

4. Actualización del sistema para medidas en *test beam*

4.1. Arquitectura del telescopio

La placa madre del sistema se ha actualizado para que pueda formar parte del sistema de adquisición de un telescopio para ser usado en un *test beam*. Un telescopio es un sistema de adquisición de trazas que se usa para medir, entre otras propiedades, la resolución espacial de nuevos detectores de silicio.

El telescopio (figura 8.1) está formado por diferentes planos paralelos (planos *xy*) con detectores de microbandas de silicio de alta resolución situados perpendiculares respecto al haz de partículas (eje *z*) para su reconstrucción. El detector bajo test (*detector under test*, DUT) se sitúa también perpendicular al haz de partículas en una posición centrada respecto a los planos de reconstrucción de trazas. Los detectores de silicio se leen en todos los planos mediante chips Beetle. Otros detectores se pueden incluir en los planos de reconstrucción de las trazas con el fin de obtener una señal de *trigger* común para el sistema de adquisición. El sistema de adquisición debe leer los datos procedentes del DUT, los datos de los detectores para reconstruir las trazas (coordenadas *x* e *y*) y los datos de los detectores de *trigger*. Para ello se utilizan las versiones modificadas de la placa madre del sistema ALIBAVA.

La resolución espacial del DUT se evalúa en este telescopio a partir de la lectura analógica de la carga recolectada en los diferentes canales del detector. El telescopio tiene una parte *front-end*, una parte de adquisición y una parte software, como se muestra en la figura 8.1. La parte *front-end* (figura 8.2) del telescopio tiene cuatro planos de medida perpendiculares al haz para la reconstrucción de las trazas y, opcionalmente, la generación de señales de *trigger* al paso del haz de

partículas. En cada uno de estos planos se sitúa una placa XYT (figuras 8.6 y 8.7). Cada placa XYT tiene dos chips Beetle para leer dos detectores de microbandas de silicio de 128 canales situados perpendiculares entre sí (*i.e.* cada uno lee las coordenadas *x* e *y* respectivamente). Cuatro pares de puntos en los planos x e y son suficientes para tener una información precisa de cada traza con suficiente resolución para analizar la respuesta espacial del DUT. Se han incluido en cada planos XY más externo un centellador conectado a un fotomultiplicador (PMT) para generar la señal de *trigger* al paso del haz de partículas. El DUT (un detector de microbandas de silicio o tipo *pixel*) se conecta a una placa DUT (figuras 8.4 y 8.5) situada entre los planos XYT (dos planos a cada lado). La placa DUT tiene dos chips Beetle para la lectura de hasta 256 canales de entrada. La placa DUT está refrigerada y se monta en una plataforma rotatoria. Así, el ángulo entre el DUT y el haz de partículas se puede variar (nominalmente es 90º). El telescopio puede tener hasta 16 planos de lectura, cuatro placas XYT y hasta doce placas DUT .

Cada placa XYT o DUT se conecta a una placa madre ALIBAVA (denominada placa esclava) actualizada para operar en el telescopio. Todas las placas esclavas (figuras 8.8 y 8.9) se conectan a una placa maestra (figura 8.10) usando un bus paralelo con un protocolo de comunicación específico. La placa maestra recibe, a través de las placas esclavas, todos los datos procedentes de los detectores y distribuye las señales de control (*Trigger, Clock* y *Event reset*) a éstas. La placa maestra se conecta al *software* de adquisición mediante *Ethernet* [125]. La parte de adquisición del telescopio se muestra en la figura 8.3. Esta parte incluye cinco placas esclavas, la placa maestra, el sistema de alimentación y el *driver* para la plataforma rotatoria de la placa DUT. Esta parte está situada en una caja específica que incluye una fuente de alimentación. La fuente de alimentación proporciona las tensiones de 5 V y 12 V necesarias para alimentar todo el *hardware* del telescopio excepto los detectores de silicio de las placas XYT y DUT. Estos detectores se alimentan con fuentes de alimentación independientes.

El *software* de adquisición recibe y procesa los datos adquiridos por el hardware para almacenarlos en un formato adecuado para su posterior análisis (reconstrucción de las trazas y evaluación del comportamiento del DUT). También se encarga de monitorizar el estado del *hardware* y controlarlo.

4.2. Actualización de la placa madre

A nivel de *hardware*, se ha sustituido el controlador USB externo en cada placa esclava por una pequeña placa con un conector para implementar un bus digital paralelo de comunicación entre las placas esclavas y placa maestra. Se mantiene el protocolo de comunicación entre la FPGA y el controlador USB.

A nivel de la lógica de la FPGA se han realizado más cambios. El diagrama de bloques de la lógica implementada en la FPGA de cada placa esclava se muestra en la figura 8.11. Se han añadido dos bloques nuevos (*Event Counter* y *External Signal Generator*) con el fin de implementar un contador de eventos adquiridos y de gestionar las señales de control común (*Trigger, Clock* y *Event reset*). otros bloques se han modificado para que puedan funcionar con dos relojes diferentes (el común y el de cada placa esclava). En la tabla 8.1 se muestran los recursos de la FPGA utilizados.

A nivel del *firmware* implementado en el procesador embebido en la FPGA se han realizado diversos cambios. El estado de calibración se ha dividido en *Calibration Acquisition* y *Calibration Reading* para facilitar la interacción con el software de adquisición. Se han implementado los estados *Data Acquisition* y *Data Reading* para la lectura y envío de los datos adquiridos de los detectores con haz de partículas.

5. Conclusiones

El diseño, desarrollo e implementación de un sistema de adquisición para detectores de silicio se ha presentado en está tesis, así como su actualización con el fin de utilizar una de este sistema, la placa madre, como parte del sistema de adquisición de un telescopio para *test beam*. Este sistema ha sido desarrollado en el marco de la colaboración ALIBAVA (Universidad de Liverpool, el CNM de Barcelona y el IFIC de Valencia), que está integrado en la colaboración RD50. La motivación principal para el diseño de este sistema surge de la necesidad de evaluar las diferentes características de los detectores de silicio, como la carga recogida o la resolución espacial, en un entorno similar al previsto para el HL-LHC, en particular, con altas dosis de radiación y una electrónica de lectura similar. Por lo tanto, en primer lugar, se ha establecido el marco en el cual se ha desarrollado este sistema.

En segundo lugar, se ha presentado una visión general de los sensores de silicio. En particular, se han resumido tanto en las propiedades del silicio como el comportamiento de unión *pn*, incluyendo las principales características de interés de la unión *pn* para los detectores de silicio. Posteriormente se ha descrito el principio de funcionamiento de los sensores de microbandas silicio. Finalmente, se han abordado otros aspectos de estos sensores como el daño de la radiación, la electrónica de lectura asociada, las fuentes de ruido o las posibles estructuras del detector.

En tercer lugar, se han detallado las motivaciones para el desarrollo de este sistema. Las especificaciones del sistema se han establecido de acuerdo a las restricciones impuestas por su aplicación. A partir de estas motivaciones y estas especificaciones, se ha presentado y discutido la arquitectura del sistema. El sistema se ha dividido en dos partes principales: una parte *hardware* y una parte *software.* También la parte *hardware* se ha basado en un sistema dual: una placa hija y una placa madre.

En cuarto lugar, la estructura y los componentes de las diferentes partes del sistema han sido descritos en detalle. Además, se ha discutido el diseño de las diferentes partes del sistema en términos de cumplimiento de los requisitos. La estructura de la placa hija y su diseño se han presentado. Esta tarjeta es una pequeña placa que contiene dos chips Beetle para la lectura de los detectores así como todo lo necesario para su conexión. También incorpora el *hardware* requerido para enviar los datos analógicos de forma segura a la placa madre así como para la recepción de las señales de control y configuración de los chips Beetle.

El diseño de la placa madre ha sido tratado en profundidad. La placa madre está destinada a procesar los datos analógicos procedentes de los chips de lectura, así como a gestionar la señal de entrada de *trigger* si se utiliza una fuente radiactiva o a generar una señal de *trigger* si se utiliza un láser. También se utiliza para controlar todo el sistema y para comunicarse con el *software* vía USB. El diagrama de bloques del *hardware* de la placa madre ha sido presentado y se ha detallado el diseño de cada bloque. A continuación, también se ha presentado el diagrama de bloques de la lógica implementada en la FPGA, describiendo el funcionamiento y la estructura de todos los bloques. Además, se han descrito la estructura y componentes del sistema embebido incluido en la FPGA. Por último, se ha explicado en detalle la funcionalidad del *firmware* programado en el procesador embebido en la FPGA. Este *firmware* ha sido implementado como una máquina de estados que incorpora todos los estados posibles del sistema.

Las principales características y la estructura de software se han presentado. La función del *software* es el control de todo el sistema y el procesamiento de los datos obtenidos de los sensores con el fin de almacenarlos en un archivo de formato adecuado. El *software* controla todo el sistema para su configuración, calibración y adquisición de datos. Incorpora una interfaz gráfica de usuario para la comunicación entre el sistema y el usuario. También genera información sobre la adquisición y almacena esta información en un formato de archivo de salida adecuado. Finalmente, el *software* debe monitorizar el estado del sistema en situaciones diferentes. El *software* se ha dividido en dos niveles según sus

funciones: un *software* de bajo nivel y un *software* de alto nivel. A bajo nivel se gestiona el intercambio de datos por USB con la placa madre y su primer procesamiento. El *software* de alto nivel implementa la interfaz gráfica de usuario , la monitorización de datos, así como la generación del archivo de salida de datos.

En quinto lugar, se ha explicado el proceso de desarrollo del sistema para validar el diseño. Por otra parte, se han presentado resultados correspondientes a medidas realizadas con el sistema. Estas medidas se realizaron tanto con un láser como con una fuente radiactiva. Se utilizaron detectores de microbandas de silicio tipo *n* y tipo *p* para analizar el rendimiento del sistema. Estas medidas muestran que el sistema funciona correctamente.

Como resultado del trabajo llevado a cabo, una serie de sistemas, 126 placas hijas y 71 placas madre, han sido producidas y testadas. El proceso de producción del sistema y los controles de calidad llevados a cabo se han explicado. Algunos sistemas (34 sistemas completos y 35 placas hija extra) se han distribuido principalmente entre los diferentes grupos de investigación de la colaboración RD50 interesados en utilizar el sistema para sus líneas de investigación. Además, otros grupos de investigación no integrados en la colaboración RD50, pero que realizan investigación con detectores silicio también han adquirido el sistema. Se han implementado en el sistema algunas mejoras basadas en la experiencia de los usuarios que se han realizado medidas con el sistema.

Por último, se ha explicado la actualización de la placa madre para utilizarla como parte del sistema de adquisición de un telescopio para *test beam*. Las motivaciones para el diseño del telescopio se han presentado. Posteriormente, se ha detallado la arquitectura del telescopio, describiendo el diseño de las diferentes partes del mismo. Se ha presentado la actualización de la lógica de la FPGA de la placa madre, y del *firmware* del procesador embebido en la misma, para su integración en el telescopio. También se han descrito las pruebas de desarrollo realizadas para validar esta actualización placa madre.

En la actualidad, el *hardware* del telescopio han sido producido y testado por separado, mientras que el *software* se está desarrollando. El siguiente paso sería terminar el desarrollo del *software* de adquisición de datos y el *software* de reconstrucción de trazas del telescopio. Posteriormente, se debe llevar a cabo la integración y la puesta en marcha del *hardware* y el *software*, así como la adquisición de datos en *test beam*.

Agradecimientos

Llegados a este punto me gustaría dedicar unas breves líneas a la gente que ha hecho posible que este trabajo haya llegado finalmente a buen puerto.

En primer lugar, me gustaría dar las gracias a Carmen García por haberme dado la oportunidad de entrar a trabajar en este campo tan apasionante de la ciencia, por su apoyo y por la confianza que siempre ha depositado en mí.

También quiero agradecer a las personas que han formado parte de la colaboración ALIBAVA, tanto del Centro Nacional de Microelectrónica de Barcelona como de la Universidad de Liverpool, la ayuda que me han prestado para sacar adelante este proyecto así como los buenos ratos que hemos pasado juntos, que no han sido pocos.

Asimismo, agradezco a muchos de los miembros de la colaboración RD50, su cálida acogida dentro de la colaboración y lo mucho que he aprendido de ellos.

Debo dar las gracias también a Vicente González, mi tutor del departamento de Ingeniería Electrónica de la Universidad de Valencia, por la amabilidad con la que siempre me ha tratado y por la ayuda que me ha prestado, desde que me matriculé en el programa de doctorado del departamento.

En el IFIC también hay mucha gente a la que tengo que agradecer su apoyo en el trabajo y su compañía diaria, las cuales me han ayudado a superar los momentos difíciles y disfrutar más de los buenos momentos. En especial quiero acordarme de Pepe, Jorge, Paco, Rosario, Rosa, José Vicente, David, Carlos E., Mercedes, Vicente, Regina y Sergio. También de toda la gente que hacen que los almuerzos y comidas en el IFIC siempre se hagan cortos.

Quiero asimismo darles las gracias a mis dos directores de tesis Salva y Carlos, por sus sabios consejos y todo el tiempo que han dedicado para que esta tesis haya sido posible. A Salva también quiero agradecerle, que además de ejercer de director, también haya sido un amigo que siempre se ha preocupado por mí y me ha apoyado de forma desinteresada.

Mi gratitud a toda mi familia y amigos más cercanos por su apoyo, cariño y los buenos momentos compartidos.

Quiero darle las gracias a Esther por estar siempre a mi lado de forma incondicional, aunque a veces se lo ponga difícil.

Por último, quiero dedicarle especialmente este trabajo a mi madre. A ella le debo todo lo que soy.

Bibliography

[1] Griffiths, David J. *Introduction to Elementary Particles*. John Wiley & Sons, Inc., 1987.

[2] CERN.*CERN in a nutshell*. CERN, 2008. Available: http://www.cern.ch.

[3] Evans, L. and Bryant, P. (editors). 'LHC Machine'. *Journal of Instrumentation*, vol. 3, no. 08, p. S08001, 2008 [Online]. Available: http://dx.doi.org/10.1088/1748-0221/3/08/S08001.

[4] The LHC Study Group. *The LHC conceptual design*. CERN/AC/95-05, CERN, 1995.

[5] The ATLAS Collaboration. 'The ATLAS Experiment at the CERN Large Hadron Collider'. *Journal of Instrumentation*, vol. 3, no. 08, p. S08003, 2008 [Online]. Available: http://dx.doi.org/10.1088/1748-0221/3/08/S08003.

[6] The CMS Collaboration. 'The CMS experiment at the CERN LHC'. *Journal of Instrumentation*, vol. 3, no. 08, p. S08004, 2008 [Online]. Available: http://dx.doi.org/10.1088/1748-0221/3/08/S08004.

[7] The ALICE Collaboration. 'The ALICE experiment at the CERN LHC'. *Journal of Instrumentation*, vol. 3, no. 08, p. S08002, 2008 [Online]. Available: http://dx.doi.org/10.1088/1748-0221/3/08/S08002.

[8] The LHCb Collaboration. 'The LHCb Detector at the LHC'. *Journal of Instrumentation*, vol. 3, no. 08, p. S08005, 2008 [Online]. Available: http://dx.doi.org/10.1088/1748-0221/3/08/S08005.

[9] The TOTEM collaboration. 'The TOTEM Experiment at the LHC'. *Journal of Instrumentation*, vol. 3, no. 08, S08007, 2008 [Online]. Available: http://dx.doi.org/10.1088/1748-0221/3/08/S08007.

[10] The LHCf collaboration. 'The LHCf Detector at the LHC'. *Journal of Instrumentation*, vol. 3, no. 08, S08006, (2008) [Online]. Available: http://dx.doi.org/10.1088/1748-0221/3/08/S08006.

[11] Gianotti, F. *et al*. 'Physics potential and experimental challenges of the LHC luminosity upgrade'. *The European Physical Journal C- Particle and Fields*, vol. 39, no. 3, p. 293-333, 2005.

[12] Ruggiero, F. *et al*. *LHC Luminosity and Energy Upgrade: A Feasibility Study*. CERN-LHC Project Report 626, CERN, 2002.

[13] RD50 Collaboration. *Development of radiation hard semiconductor devices for very high luminosity colliders. RD50 R&D Proposal*. CERN-

LHCC-2002-003, CERN, 2002.
[14] Sze, S.M. *Physics of Semiconductor Devices*. John Wiley & Sons, Inc., 1981.
[15] Georgia State University. *Hyperphysics. Energy bands: insulators and semiconductors*. Department of Physics and Astronomy. Georgia State University, 2005. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/hframe.html
[16] Lutz, G. *Semiconductor Radiation Detectors: Device Physics*. Springer-Verlag, 1999.
[17] Spieler, H. *Semiconductor detector systems*. Oxford University Press, 2005.
[18] Leo, W.R. *Techniques for nuclear and particle physics experiments: a how-to approach*. Springer-Verlag, 1994.
[19] Landau, L. 'On the energy loss of fast particles by ionization'. *J. Phys.USSR,* vol. 8, pp. 201-205, 1944.
[20] Ramo, S. 'Currents induced by electron motion'. *Proceedings of the IRE*, vol. 27, p. 584, 1939.
[21] Casse, G. *The effect of hadron irradiation on the electrical properties of particle detectors made from various silicon materials*. PhD Thesis, University of Joseph Fourier, 1998.
[22] Moll, M. *Radiation damage in silicon particle detectors - microscopic defects and macroscopic properties*. PhD Thesis, University of Hamburg, 1999.
[23] Sievers, P. *A Silicon Inner Tracker for the LHCb Experiment.* PhD. Thesis, University of Zürich, 2002.
[24] Sonnenblick, R. *et al*. 'Electrostatic simulations for the design of silicon strip detectors and front-end electronics'. *Nuclear Instruments and Methods A,* vol. 310, pp.189-191, 1991.
[25] Messenger, G. C. 'Collection of charge on junction nodes from ion tracks'. *IEEE Transactions on Nuclear Science*, vol. NS-29, No. 6, pp. 2024-2031, 1982.
[26] Gatti, E. and Manfredi, P. F. 'Processing the signals from solid-state detectors in elementary-particle physics'. *Rivista del Nuovo Cimento,* vol. 9, no.1, pp.1-146, 1986.
[27] Pallás-Areny, R. and Webster, J. G. *Analog signal processing*. John Wiley & Sons, Inc., 1999.
[28] Spieler, H. *Radiation Detectors and Signal Processing.* Lecture Notes, Heidelberg, 2001. [Online]. Available: http://www-physics.lbl.gov/~spieler/Heidelberg_Notes/.
[29] Kramberger, G. *Signal development in irradiated silicon detectors*. PhD Thesis, University of Ljubljana, 2001.

[30] Bates, S. J. *The effects of Proton and Neutron Irradiations on Silicon Detectors for the LHC*. PhD Thesis, University of Cambridge, 1993.
[31] RD50 Collaboration. *Radiation hard semiconductor devices for very high luminosity colliders. RD50 Status Report 2006.* CERN-LHCC-2007-005, CERN, 2007.
[32] Knoll, G. F. *Radiation Detection and Measurements*. John Wiley & Sons, Inc., 2000.
[33] Kodys, P. *et al.* 'Laser tests of silicon detectors'. *Nuclear Instruments and Methods A,* vol. 573, pp. 12-15, 2007.
[34] Escobar, C. *Sistema láser para el estudio de los módulos del SCT de ATLAS*. Master Thesis, Univerity of Valencia, 2005.
[35] Menke, S. *et al.* 'Performance of a beam telescope using double sided silicon microstrip detectors'. *Nuclear Instruments and Methods A,* vol. 364, pp. 224-230, 1995.
[36] Djama, F. *et al.* 'A submicron precision silicon telescope for beam test purposes'. *Nuclear Instruments and Methods A,* vol. 372, pp. 379-384, 1996.
[37] de Palma, M. *et al.* 'A high resolution beam telescope built with double sided silicon strip detectors'. *Nuclear Instruments and Methods A,* vol. 381, pp. 49-56, 1996.
[38] Mäenpää, T. *et al.* 'Silicon beam telescope for LHC upgrade tests'. *Nuclear Instruments and Methods A,* vol. 593, pp. 523-529, 2008.
[39] Amsler, C. *et al.* 'A high resolution silicon beam telescope'. *Nuclear Instruments and Methods A,* vol. 480, pp. 501-507, 2002.
[40] Treis, J. *et al.* 'A modular PC based silicon micrsotrip beam telescope with high speed data acquisition'. *Nuclear Instruments and Methods A,* vol. 490, pp. 112-123, 2002.
[41] Löchner, S., Schmelling, M. *The Beetle Reference Manual: chip version 1.3, 1.4 and 1.5*. CERN-LHCb-2005-105, CERN, 2006.
[42] Philips semiconductors. *The I2C-bus specification and user manual. Version 3.0.* UM10204, 2007.
[43] National Semiconductor. *LVDS Owner's Manual.* 550062-004, 2008.
[44] Löchner, S. *Development, Optimisation and Characterisation of a Radiation Hard Mixed-Signal Readout Chip for LHCb.* PhD. Thesis, University of Heidelberg, 2006.
[45] Bausmeister, D. *Development and Characterisation of a Radiation Hard Readout Chip for the LHCb-Experiment*, PhD. Thesis, University of Heidelberg, 2003.
[46] Sexauer, E. *Development of Radiation Hard Readout Electronics for LHCb*. PhD. Thesis, University of Heidelberg, 2001.
[47] Johnson, H., Graham, M. *High-speed digital design*. Prentice Hall, 1993.

[48] Montrose, M. *EMC and the Printed Circuit Board*. IEEE Press, 1999.
[49] Ott, H.W. *Noise Reduction Techniques in Electronic Systems*. John Wiley & Sons, 1988.
[50] Analog Devices. *AD8132 data sheet*. 2009.
[51] Rolli, K., Buytaert, J. *40 Ms/s Analog transmission on Shielded Twisted Pair cable*. LHCb-98-032 Note, 1998.
[52] Huq, S., Goldie, J. *An overview of LVDS technology*. National Semiconductor, AN-971, 1998.
[53] National Semiconductor. *LM1117 data sheet*. 2006.
[54] Texas Instruments. *LP2985 data sheet*. 2004.
[55] Analog Devices. *AD8129/8130 data sheet.* 2005.
[56] National Semiconductor. *LMH6559 data sheet*. 2006.
[57] Analog Devices. *AD8139 data sheet*. 2004.
[58] Maxim-Dallas. *MAX1448 data sheet*. 2004.
[59] Analog Devices. *Fundamentals of sampled data systems*. AN-282, 1991.
[60] Shannon, C. E. 'Communication in the presence of noise'. *Proceedings Institute of Radio Engineers*, vol. 37, no.1, pp. 10-21, Jan. 1949.
[61] National Semiconductor. *DS90LV047A data sheet*. 2003.
[62] TDK. *ZJYS51R5-4PT-01 data sheet*. 2006.
[63] Pulse Engineering. *Understanding Common Mode Noise.* G019, 1999.
[64] National Semiconductor. *DS90LV048A data sheet*. 2001.
[65] U.S. NIM Committee. *Standard NIM Instrumentation System.* DOE/ER-0457T. 1990.
[66] Maxim-Dallas. *MAX9600-MAX9602 data sheet*. 2002.
[67] Holland, N. *Interfacing between LVPECL,VML, CML and LVDS levels*. Texas Instruments, Application Report SLLA120, 2002.
[68] Shockman, P. *Termination of ECL devices with emitter follower output structure*. ON Semiconductor, Application Note AN8020/D, 2004.
[69] Cadence PSpice A/D Home Page. Available: http://www.cadence.com/products/orcad/pspice_simulation/pages/default.aspx
[70] Burr-Brown. *DAC7614 data sheet*. 1998.
[71] Acam mess-electronic. *TDC-GP1 data sheet*. 2001.
[72] Data Delay Devices. *3D3428 data sheet*. 2004.
[73] Horowitz, P. and Hill, W. *The art of electronics*. Cambridge University Press, 1989.
[74] Philips Semiconductors. *BSR13-BSR14 NPN switching transistors data sheet*. 1999.
[75] Maxim-Dallas. *MAX6682 data sheet*. 2002.
[76] Wakerly, J. F. *Digital Design: Principles and Practices.* Prentice-Hall, 2000.

[77] Micron Technology. *256 Mb SDR SDRAM data sheet.* 1999.
[78] USB Implementers Forum. *Universal Serial Bus Specification*. Revision 2.0., 2000.
[79] USB Implementers Forum. *Universal Serial Bus 3.0 Specification*. 2008.
[80] FTDI. *FT245R USB FIFO IC data sheet*. 2005.
[81] Cypress Semiconductor. *EZ-USB FX2LP™ USB Microcontroller data sheet.* 2005.
[82] Microchip Technology. *PIC18F/LF1XK50 Data Sheet. 20-Pin USB Flash Microcontrollers with nanoWatt XLP Technology.*DS41350E, 2010.
[83] Brown, S. and Rose, S. 'Architecture of FPGAs and CPLDs: A Tutorial'. *IEEE Design and Test of Computers*, Vol. 13, No. 2, pp. 42-57, 1996.
[84] Zeidman, B. *Designing with FPGAs and CPLDs*. CMP Books, 2002.
[85] Chan, P. and Murrad, S. *Digital Design using Field-Programmable Gate Arrays*. Prentice-Hall, 1994.
[86] Oldfield J. and Dorf R. *Field-Programmable Gate Arrays. Reconfigurable Logic for Rapid Prototyping and Implementation of Digital Systems*. John Wiley & Sons, Inc., 1995.
[87] IEC/IEEE. *Behavioural languages. Part 1-1:VHDL language reference manual*. IEC 61691-1-1, IEEE Std.1076, 2004.
[88] IEEE. *IEEE Standard for Verilog Hardware Description Language.* IEEE Std. 1364, 2005.
[89] Xilinx. *Spartan-3 FPGA family data sheet.* DS099, 2005.
[90] Alexander, M. *Power Distribution System (PDS) Design: Using Bypass/Decoupling Capacitors*. Xilinx, XAPP623, 2005.
[91] Xilinx. *Platform Flash In-System Programmable Configuration PROMs*. DS123, 2005.
[92] Murata. *Block type EMI filter LC combined type, BNX02 series data sheet*. 2006.
[93] Traco Power. *TEL3 series 3 Watt DC/DC converters data sheet*. 2005.
[94] Sclocchi, M. *Input Filter Design for Switching Power Supplies.* National Semiconductor, 2002.
[95] Ericsson. *Output Filter Design (for DC/DC converters)*. Design Note 011, 2005.
[96] C&D Technologies. *DC-DC Converters Application Notes*. AN006.1, 2000.
[97] C&D Technologies. *Isolated 5W Regulated Single & Dual Output DC/DC Converters, NMXSO series data sheet*. KDC_NMXSOC.1. 2006.
[98] C&D Technologies. *Non-isolated DC/DC Converters, NNL05 series data sheet*. KDC_NNL05C.5, 2005.
[99] National Semiconductor. *LP38691/LP38693 data sheet*. 2005.
[100] Mears, J. A. *Transmission line operation and applications guide*. National

Semiconductor, Application note 905, 1996.
[101] Xilinx Inc. *MicroBlaze Processor Reference Guide.* UG081, October 2005.
[102] Herveille, R. *I2C Master Core Specification.* Rev. 0.9, July 2003. Available: http:// www.opencores.org.
[103] OpenCores Organization. *WISHBONE System-On-Chip Interconnection Architecture Specification*. Rev. B.3, September 2002. Available: http:// www.opencores.org.
[104] Xilinx Inc. *On-Chip Peripheral Bus V2.0 with OPB Arbiter.* DS401, June 2005.
[105] Xilinx Inc. *OPB SDRAM Controller.* DS426, May 2005.
[106] Xilinx Inc. *Fast Simplex Link (FSL) Bus*. DS449, April 2005.
[107] Xilinx Inc. *Using Digital Clock Managers (DCMs) in Spartan-3 FPGAs*. XAPP462, January 2006.
[108] International Organization for Standarization (ISO). *C programming language standard.* ISO/IEC 9899:1999:TC3, September 2007.
[109] The ROOT Team. *ROOT Users Guide 5.26.* December, 2009. Available: http://root.cern.ch/
[110] International Organization for Standarization (ISO). *C++ programming language standard*. ISO/IEC 14882:2003. September 2007.
[111] Lacasta, C. *Alibava-gui documentation*. November, 2009. Available: https://twiki.ific.uv.es/twiki/bin/view/Atlas/ALiBaVa
[112] The Mathworks. Using MATLAB. Version 6.5 (Release 13). July, 2002. Available: http://www.mathworks.com/
[113] Miñano, M. *et al*. 'Characterization of irradiated P-type silicon detectors by the ALIBAVA system'. *Nuclear Instruments and Methods A,* vol. 617, pp. 565-567, 2010.
[114] Soldevila, U. *et al.* 'Neutron irradiation for p-type sensors. Detector characterization with ALIBAVA system'. *14th RD50-Workshop on Radiation hard semiconductor devices for very high luminosity colliders,* 3-5 June 2009.
[115] Xilinx Inc. *Platform Cable USB*. DS300, June 2006.
[116] Pacifico, N. *et al.* 'Annealing of Charge Collection Efficiency in highly irradiated low-resistivity MCz-p-on-n strip detectors'. *15th RD50-Workshop on Radiation hard semiconductor devices for very high luminosity colliders,* 16-18 November 2009.
[117] Walz, M. *et al.* 'Charge Collection Measurements on Irradiated Planar Silicon Strip Sensors'. *16th RD50-Workshop on Radiation hard semiconductor devices for very high luminosity colliders,* 31-2 June 2010.
[118] Lucas, C. *et al.* 'Charge Collection Study of Heavily Irradiated Silicon Microstrip Detectors'. *17th RD50-Workshop on Radiation hard*

semiconductor devices for very high luminosity colliders, 17-18 November 2010.

[119] Eber, R. *et al.* 'Annealing study of a highly irradiated FZ CMS mini sensor with the ALiBaVa setup at KIT'. *17th RD50-Workshop on Radiation hard semiconductor devices for very high luminosity colliders,* 17-18 November 2010.

[120] Weigell, P. *et al.* 'Characterization of 75µm and 150µm Thin Strip and Pixel Sensors Produced at MPP-HLL'. *16th RD50-Workshop on Radiation hard semiconductor devices for very high luminosity colliders,* 31-2 June 2010.

[121] Köhler, M. *et al.* 'Test Beam and Laser Measurements With Irradiated 3D Silicon Strip Detectors'. *16th RD50-Workshop on Radiation hard semiconductor devices for very high luminosity colliders,* 31-2 June 2010.

[122] Bates, R. *et al.* 'Irradiation studies of CNM double sided 3D detectors'. *15th RD50-Workshop on Radiation hard semiconductor devices for very high luminosity colliders,* 16-18 November 2009.

[123] Vila, I. *et al.* 'A novel two-dimensional microstrip sensor for charge division readout'. *17th RD50-Workshop on Radiation hard semiconductor devices for very high luminosity colliders,* 17-18 November 2010.

[124] Sadrozinski, H. and Seiden, A. 'Tracking detectors for the sLHC, the LHC upgrade'. *Nuclear Instruments and Methods A,* vol. 541, pp. 434-440, 2005.

[125] IEEE. *IEEE Standard for Information technology-Specific requirements. Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.* IEEE 802.3, 2008.

[126] Xilinx. *Spartan-3E FPGA Starter Kit Board User Guide.* UG230, 2008.